

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA DE TRANSPORTES E LOGSTICA

LARISSA BAGINI BARBOSA

**OTIMIZAÇÃO DO SEQUENCIAMENTO DE TAREFAS EM MÁQUINAS
PARALELAS COM TEMPOS DE PROCESSAMENTO DIFERENTES**

Joinville

2017

LARISSA BAGINI BARBOSA

**OTIMIZAÇÃO DO SEQUENCIAMENTO DE TAREFAS EM MÁQUINAS
PARALELAS COM TEMPOS DE PROCESSAMENTO DIFERENTES**

Trabalho de conclusão de curso apresentado como requisito parcial para obtenção do título de bacharel em Engenharia de Transportes e Logística, curso de Engenharia de Transportes e Logística, Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Orientador: Prof.^a Dra. Silvia Lopes de Sena Taglialenha.

Joinville

2017

OTIMIZAÇÃO DO SEQUENCIAMENTO DE TAREFAS EM MÁQUINAS PARALELAS COM TEMPOS DE PROCESSAMENTO DIFERENTES

Este trabalho de conclusão de curso foi julgado adequado para obtenção do Título de Graduado e aprovado em sua forma final pelo curso de Engenharia de Transportes e Logística.

Joinville, 01 de dezembro de 2017.

Prof^a. Dra. Elisete Santos da Silva Zagheni
Coordenador do Curso

Banca examinadora:

Prof^a. Dra. Silvia Lopes de Sena Tagliapietra
Orientadora
Universidade Federal de Santa Catarina

Prof^a. Dra. Christiane Wenck Nogueira
Universidade Federal de Santa Catarina

Prof^a. Dra. Vanina Macowski Durski Silva
Universidade Federal de Santa Catarina

AGRADECIMENTOS

A Deus que me deu a graça da vida e pela oportunidade de me desenvolver e constantemente aprender.

A minha família por me proporcionarem uma educação que me permitiu alcançar meus objetivos e realizações, e por terem me apoiado em todos os momentos para chegar até aqui.

A minha orientadora, pelo grande auxílio e atenção no desenvolvimento deste trabalho e por ser uma profissional a se espelhar.

Aos meus amigos e colegas de graduação pelo auxílio e paciência durante todo o processo da graduação e execução deste trabalho.

“O insucesso é apenas uma oportunidade de recomeçar de novo com mais inteligência ”

Henry Ford.

RESUMO

Em um ambiente econômico cada vez mais competitivo e concorrido, as indústrias necessitam ser cada vez mais eficientes para se manterem no mercado. A utilização de soluções em otimização vem se mostrando uma ótima alternativa para reduzir custos de produção sem perder a qualidade do produto final entregue ao cliente. Este trabalho tem seu foco no problema de otimização do sequenciamento de tarefas em máquinas paralelas com diferentes tempos de processamento e com penalidade por atraso de produção, baseado em um caso real numa indústria do setor de injeção plástica. Para resolução do problema propõe-se um modelo de programação linear inteira mista e apresenta-se uma forma de resolução exata do modelo proposto através de *solvers* processados remotamente. Para a elaboração do modelo definiu-se as restrições considerando-se as particularidades do sistema produtivo e preceitos estabelecidos de forma a otimizar o sequenciamento. Para demonstrar a validação do modelo proposto considerou-se sua aplicação em cenários estabelecidos com solução ótima conhecida previamente. Iniciou-se os testes com duas máquinas e sete tarefas e expandiu-se esse número até não se conseguir processar o modelo. Os resultados obtidos quando se aplicou os dados reais da empresa para até 24 máquinas e 59 tarefas demonstrou a possibilidade de melhorar a utilização dos recursos disponíveis e redução da multa por atraso no prazo de produção em 11%. Essa redução é economicamente considerável, uma vez que além de aumentar os lucros da empresa, também possibilita uma melhora no nível do serviço entregue ao cliente. Academicamente, tem-se a contribuição de uma modelagem que garante a solução exata para um número de máquinas e tarefas em que, na literatura, só se encontram métodos de resolução aproximados.

Palavras-chave: Scheduling. Sequenciamento de tarefas em máquinas paralelas. Modelo exato para sequenciamento de tarefas. Otimização da produção.

ABSTRACT

In an increasingly competitive economic environment, industries need to be efficient to stay in the market. The use of solutions in optimization has been proved a good alternative to reduce costs without losing the quality of the final product delivered to the customer. This work focuses on the problem of task sequencing optimization in parallel machines with different processing times and with delayed production penalty, based on a real case in a plastic injection industry. To solve the problem is proposed a mixed integer linear programming model and an exact solution method through solvers processed remotely. For the model elaboration, it was defined restrictions considering the particularities of the productive system and established precepts, in order to optimize the sequencing. To demonstrate the validation of the proposed model, it was considered its application in scenarios established with optimal solution previously known. The tests started with two machines and seven tasks and this number was expanded until the model could not be processed. The results obtained when applying the actual data of the company, for up to 24 machines and 59 tasks, showed the possibility of improving the use of available resources and reduction of the penalty by delay in production term by 11%. This reduction is economically considerable, since in addition to increasing the company's profits, it also enables an improvement in the level of service delivered to the customer. Academically, the contribution was a modeling that guarantees the exact solution for a number of machines and tasks in which, in the literature, only approximate resolution methods are found.

Keywords: Scheduling. Parallel machines sequencing of tasks on. Task sequencing exact model. Production optimization.

LISTA DE FIGURAS

Figura 1	Etapas da metodologia.....	19
Figura 2	PPCP: base para a competitividade empresarial.....	25
Figura 3	Exemplo de processamento em máquina única.....	31
Figura 4	Exemplo de processamento em máquinas paralelas.....	32
Figura 5	Exemplo de processamento em máquinas em <i>Job Shop</i>	33
Figura 6	Exemplo de processamento em máquinas em <i>Flow Shop</i>	34
Figura 7	Exemplo de processamento em máquinas em <i>Flexible Flow Shop</i>	35
Figura 8	Modelo em AMPL.....	44
Figura 9	Arquivo para entrada de dados.....	45
Figura 10	Sequência de tarefas em uma máquina.....	45
Figura 11	Modelo generalizado em AMPL.....	55
Figura 12	Modelo generalizado em AMPL arquivo de dados.....	56
Figura 13	Página inicial do site.....	57
Figura 14	Tipos de problemas disponíveis no NEOS Server.....	58
Figura 15	Alguns <i>solvers</i> disponíveis NEOS Server.....	59
Figura 16	<i>Input</i> dos dados.....	60
Figura 17	Sequenciamento máquinas.....	62
Figura 18	Modelo generalizado em AMPL cenário 1.....	63
Figura 19	Sequenciamento máquinas cenário 1.....	64
Figura 20	Modelo generalizado em AMPL cenário 2.....	65
Figura 21	Sequenciamento máquinas cenário 2.....	66
Figura 22	Modelo generalizado em AMPL cenário 3.....	67
Figura 23	Sequenciamento máquinas cenário 3.....	68
Figura 24	Sequenciamento para 24 máquinas.....	71
Figura 25	Sequencia atual.....	74

LISTA DE TABELAS

Tabela 1	Dados para sequenciamento em máquina única.....	40
Tabela 2	Tempo de processamento para duas máquinas.....	54
Tabela 3	Tempo de processamento para duas máquinas cenário 1.....	62
Tabela 4	Tempo de processamento para duas máquinas cenário 2.....	64
Tabela 5	Tempo de processamento para duas máquinas cenário 3.....	66
Tabela 6	Quantidade de itens por tarefa.....	69
Tabela 7	Tempo de processamento para o modelo para 24 máquinas e 59 tarefas.....	86
Tabela 8	Prazo e multa para 24 máquinas e 59 tarefas.....	87

LISTA DE GRÁFICOS

Gráfico 1	Multa: cenário real e resultado obtido.....	75
Gráfico 2	Número de iterações	76
Gráfico 3	Número de variáveis	77
Gráfico 4	Número de restrições	77

LISTA DE SIGLAS E ABREVIATURAS

PO: Pesquisa Operacional

PL: Programação Linear

PLI: Programação Linear Inteira

PLIM: Programação Linear Inteira Mista

PPCP: Planejamento, programação e controle de produção

MTO: Produção por ordem de venda, do inglês *Make-to-Order*

MTS: Produção para estocagem, do inglês *Make-to-Stock*

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	OBJETIVOS DO TRABALHO	16
1.1.1	Objetivo Geral	16
1.1.2	Objetivos Específicos	16
1.2	JUSTIFICATIVAS	17
1.3	METODOLOGIA E ESTRUTURA DO TRABALHO	18
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	PROGRAMAÇÃO LINEAR INTEIRA	20
2.2	PLANEJAMENTO, PROGRAMAÇÃO E CONTROLE DE PRODUÇÃO.....	22
2.2.1	Objetivos do PPCP	25
2.2.2	Etapas do PPCP	27
2.3	SEQUENCIAMENTO DA PRODUÇÃO	28
2.3.1	Sequenciamento de Tarefas	29
2.3.2	Sequenciamento em Máquinas	30
2.4	MÉTODOS DE SOLUÇÃO	35
2.4.1	Métodos Heurísticos	36
2.4.2	Método exato	38
3	MODELO MATEMÁTICO PARA SEQUENCIAMENTO EM MÁQUINA.....	40
3.1	MÁQUINA ÚNICA.....	40
3.1.1	Restrições para o sequenciamento em uma máquina	41
3.1.2	Modelo em AMPL para o sequenciamento em uma máquina.....	44
3.2	SEQUENCIAMENTO DE TAREFAS EM VÁRIAS MÁQUINAS.....	45
3.3	PREMISSAS PARA O PROBLEMA.....	47
3.4	MODELAGEM PROPOSTA.....	49
3.4.1	Definição das variáveis	49

3.4.2	Restrições de não interferência entre as tarefas	50
3.4.3	Restrições de cobertura	51
3.4.4	Restrições de empacotamento	51
3.4.5	Restrições de cumprimento do prazo de execução	51
3.4.6	Multa por atraso	53
3.4.7	Modelo algébrico considerando n tarefas e m máquinas	53
3.5	ESTRATÉGIA DE SOLUÇÃO	56
3.6	VALIDAÇÃO DO MODELO	61
3.6.1	Cenário 1	62
3.6.2	Cenário 2	64
3.6.3	Cenário 3	66
3.7	APLICANDO O MODELO PARA $m = 24$ MÁQUINAS E $n = 59$ TAREFAS	68
4	RESULTADOS E ANÁLISES COMPARATIVAS	73
4.1	CENÁRIO ATUAL	73
4.2	DESEMPENHO DO SEQUENCIAMENTO	75
4.3	ANÁLISE DA COMPLEXIDADE DO MODELO	75
5	CONSIDERAÇÕES FINAIS	79
5.1	DESDOBRAMENTOS E TRABALHOS FUTUROS	80
	REFERÊNCIAS BIBLIOGRÁFICAS	81
	ANEXO A – DADOS	86

1 INTRODUÇÃO

Para alcançar uma melhoria no nível de serviço de atendimento aos seus clientes, é natural que empresas procurem a otimização dos processos da cadeia produtiva, a fim de atingir um nível de excelência em suas operações. Variadas técnicas de otimização têm sido desenvolvidas e empregadas em indústrias com o intuito de reduzir ao máximo os custos de produção ou até mesmo aumentar o rendimento de processos produtivos (BOWERSOX, 2011).

O planejamento de tarefas (*scheduling*) é uma etapa economicamente importante para diversos setores produtivos, tais como: produção em indústrias de manufatura, transporte de cargas, processos computacionais em sistemas operacionais, escala de tripulações, embarque de carga em navios, coleta e distribuição de produtos e etc.

Problemas de otimização combinatória surgem quando existe a necessidade de selecionar, a partir de um conjunto discreto e finito de dados, o melhor subconjunto que satisfaça os critérios definidos anteriormente (GOLDBARG E LUNA, 2005).

Problemas de planejamento são problemas de otimização tipicamente combinatórios que pertencem à classe de problemas *NP-hard* (problemas que não podem ser resolvidos de forma determinística em tempo polinomial).

Para Luh (1998) no que diz respeito à eficiência produtiva, o sequenciamento é um fator chave, já que está ligado ao planejamento estratégico da empresa, por ser capaz de reduzir estoques, melhorar datas de entrega e melhorar gargalos do sistema. Segundo Blazewicz *et al.* (1996), as duas restrições clássicas dos problemas de sequenciamento são que cada tarefa deve ser processada por uma única máquina e que cada máquina é capaz de processar uma única tarefa por vez.

Entre os principais tipos de problemas de programação da produção encontra-se o problema de sequenciamento em máquinas com penalidade por atraso na produção (GOMES JÚNIOR, 2007); e sua resolução implica em redução de custos gerados com atraso. Esse atraso na produção gera atraso na entrega do produto final ao cliente acarretando em uma possível perda do mesmo.

Existem duas classes de métodos de resolução para os problemas de sequenciamento: os métodos exatos e os métodos heurísticos. Embora os métodos heurísticos apresentem a característica de demandarem baixo tempo de execução, em geral não há garantia de se alcançar a solução ótima. Já os métodos exatos, em contrapartida, determinam a solução ótima, mas em geral, demandam tempo de execução muito alto, além disso em poucos estudos sobre métodos de otimização para o problema de sequenciamento de tarefas em máquinas consideram essa abordagem.

No entanto, esta premissa está mudando, uma vez que já é possível resolver problemas de grande porte com a utilização remota de *softwares* especializados, como por exemplo, através do *site neos-server.org Server* (NEOS GUIDE SERVER, 2017), processados em supercomputadores, como será melhor explicado na seção 3.4 deste trabalho.

Neste trabalho, será considerado um problema real de sequenciamento de tarefas em máquinas em uma indústria de conexões plásticas, através da abordagem exata de solução. Será apresentado um modelo matemático que possibilitará resolver o problema de sequenciamento de tarefas de uma indústria multinacional brasileira do setor plástico com sede em diversos países.

Os dados considerados foram obtidos de uma filial sediada nos Estados Unidos com 24 máquinas onde se faz a injeção plástica, onde, cada uma dessas máquinas produz uma série de tarefas e as tarefas podem ser realizadas por mais de uma máquina.

1.1 OBJETIVOS DO TRABALHO

Nesta seção apresentam-se os objetivos geral e específicos que se pretende alcançar neste trabalho.

1.1.1 Objetivo Geral

Propor e implementar um modelo matemático exato, atendendo às restrições operacionais de uma indústria do setor plástico, para encontrar o melhor sequenciamento de tarefas em máquinas.

1.1.2 Objetivos Específicos

Para alcançar o objetivo geral proposto neste trabalho será necessário também atingir os seguintes objetivos específicos:

- Aprofundar conhecimentos sobre o tema sequenciamento de tarefas em máquinas, evidenciando definições, variedades e possíveis abordagens;
- Apresentar um diagnóstico da atual situação da fábrica de conexões plásticas;
- Avaliar a gestão da produção da fábrica com uso de sequenciamento de tarefas;
- Determinar equações matemáticas que descrevam as restrições impostas pela empresa através de um modelo matemático exato;
- Determinar através de ferramentas computacionais soluções viáveis para o problema.

1.2 JUSTIFICATIVAS

O problema a ser considerado neste trabalho refere-se à uma indústria multinacional do setor plástico, com planta fabril localizada nos Estados Unidos. Durante a realização de estágio na planta de Joinville foi possível acompanhar os processos produtivos da planta americana e analisar várias de suas etapas.

Para Spak (2012) com mercados cada vez mais competitivos surge a necessidade de as organizações apresentarem maior precisão e eficácia em suas tomadas de decisão. Para tanto, torna-se necessário a utilização de ferramentas estruturadas para suporte a estas decisões. Estas ferramentas necessitam de viabilidade em sua utilização. A aplicação de métodos e modelos matemáticos que reduzam custos, de maneira que não impactem na qualidade do produto final entregue ao cliente, é de grande interesse das companhias (BALLOU, 2006).

Determinar a melhor programação da produção tem sido um objetivo a ser alcançado por muitas empresas tendo em vista que, com isso, consegue-se a redução dos custos de produção, com reduções de multas por atraso de entrega, e/ou necessidade de estocagem, pela possibilidade de evitar-se produzir antes da necessidade (GOMES JÚNIOR, 2007).

Neste sentido a contribuição econômica desse trabalho se dá:

- À empresa, por encontrar a sequência ótima de processamento das tarefas nas máquinas reduzindo a multa paga pela fábrica por atraso de produção e, portanto, aumentando seus lucros;
- E aos clientes, pois possibilita melhorar o nível de serviço entregue.

A literatura disponível para o problema de sequenciamento de tarefas em máquinas não é muito ampla quando se trata da abordagem exata. Na pouca literatura encontrada seguindo essa metodologia há uma restrição quanto ao número de máquinas, que na maioria considera apenas uma ou duas máquinas e um número reduzido de tarefas. Sendo assim, a contribuição acadêmica do trabalho está em desenvolver um modelo matemático inédito para a resolução do problema de

sequenciamento de tarefas, além de sequenciar o problema para um número grande máquinas.

Neste contexto, o trabalho justifica-se ao analisar o processo produtivo de uma indústria do setor plástico, com restrições de capacidade e busca uma solução otimizada para o problema de sequenciamento de tarefas nas máquinas utilizando abordagem exata.

1.3 METODOLOGIA E ESTRUTURA DO TRABALHO

Almejando tornar possível a obtenção dos objetivos propostos neste trabalho, organizou-se uma metodologia de trabalho que por sua finalidade pode ser considerada uma pesquisa aplicada que, de acordo com Gil (2008) tem como característica o interesse na aplicação, utilização e consequências práticas dos conhecimentos. A Figura 1 apresenta as principais etapas da metodologia proposta.

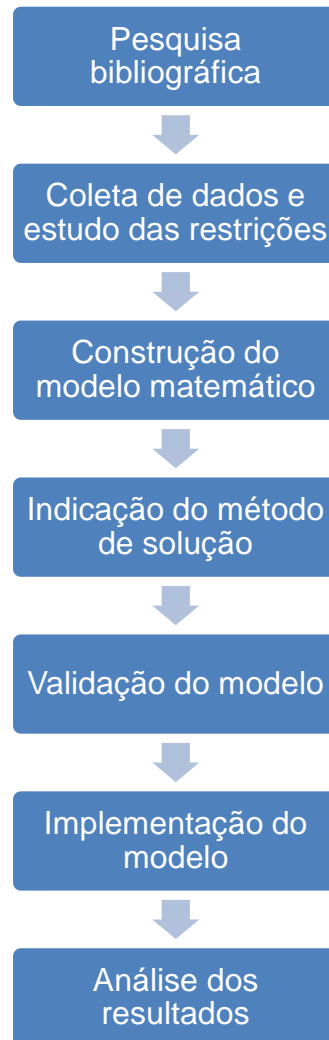
Os conceitos estão dispostos de forma a representar a cronologia em que serão utilizados ou consideradas para melhor compreensão do trabalho

Inicialmente, é realizada a pesquisa bibliográfica sobre os temas que construíram o referencial teórico do trabalho, como por exemplo, programação linear inteira, modelos de otimização, planejamento, programação e controle da produção, escalonamento da produção, sequenciamento de tarefas e de máquinas.

Em seguida, ocorre a coleta de dados e análise das restrições a serem consideradas que serão utilizados como base para as etapas seguintes, que consistem na construção do modelo matemático e na identificação do método de resolução a ser implementado. Há também a validação do modelo matemático

Na etapa seguinte ocorre a implementação do modelo em linguagem de programação AMPL (*A Mathematical Programming Language*) utilizando o *solver* GUROBI na plataforma Neos Server que será melhor detalhada na seção 3.3 deste trabalho. Por fim analisa-se dos resultados obtidos.

Figura 1 – Etapas da metodologia



Fonte: Autoria Própria

O trabalho propõe uma nova abordagem matemática para o problema do sequenciamento de tarefas em máquinas aplicada em uma indústria de injeção plástica que produz conexões localizada nos Estados Unidos da América. No próximo capítulo inicia-se a descrição da etapa 1.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados os principais conceitos relacionados ao tema do trabalho, buscando gerar embasamento teórico a respeito da pesquisa operacional, do planejamento e programação de produção, bem como do método utilizado para o desenvolvimento do estudo de caso.

2.1 PROGRAMAÇÃO LINEAR INTEIRA

Segundo Goldberg e Luna (2005) a programação linear é o modelo básico para a compreensão de todos os outros modelos existentes de programação matemática e seus conceitos serão extensíveis aos demais, servindo de suporte a estudos mais avançados. Além disso, é considerado extremamente eficiente nos algoritmos existentes, por ter alta capacidade de cálculo e podendo ser facilmente implementados por meio de planilhas, como por exemplo no *software Excel*.

A PL visa fundamentalmente encontrar a melhor solução possível para problemas com funções lineares no modelo. Sua tarefa consiste em maximizar ou minimizar uma função linear, denominada função objetivo, respeitando as restrições de igualdade ou desigualdade também lineares (MARINS, 2011).

O modelo matemático básico de um problema de PL pode ser representado pelas equações abaixo, formulado por Hillier e Lieberman (2013).

$$\text{Maximizar } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (2.1)$$

Sujeito às restrições:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \quad (2.2)$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \quad (2.3)$$

.

.

.

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \quad (2.4)$$

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0 \quad (2.5)$$

Em que a equação (2.1) representa o objetivo a ser otimizado (minimização ou maximização), as restrições (2.2) - (2.4) genéricas representam as restrições operacionais ou de capacidade a serem cumpridas, e as equações (2-5) definem o tipo de variáveis a serem consideradas.

Segundo Hillier e Lieberman (2013) para muitos problemas práticos as variáveis de decisões somente fazem sentido se possuírem valores inteiros. O modelo matemático de problemas de programação linear inteira (PLI) pode ser representado pelo modelo (2.1) - (2.5), acrescido da restrição de que as variáveis devem ter valores inteiros, ou seja, a restrição (2.6) também deverá ser considerada no modelo:

$$x_1, x_2, \dots, x_n \in \mathbb{Z}. \quad (2.6)$$

Caso apenas algumas das variáveis tenham valores inteiros trata-se de um modelo denominado programação linear inteira mista (PLIM) (ARENALES, 2006).

A programação linear inteira (PLI) é um ramo da PO que aborda métodos de otimização para problemas modelados através de um modelo linear inteiro, em que as variáveis pertencem ao conjunto dos números inteiros (ou discretos) (TAHA, 2008).

Segundo Colin (2013), a PLI teve origem em 1957, quando Ralph Gomory trabalhava como consultor para a Marinha dos Estados Unidos da América, em que um dos participantes da equipe que apresentava um modelo de programação linear, comentou sobre o quão interessante seria se as respostas viessem em números inteiros, levando em conta que uma resposta do tipo “1,5 avião de carga” não significa nada.

O mesmo autor menciona que Gomory chegou então a um raciocínio que levava dois pontos em consideração:

- Se a programação linear encontra soluções ótimas que podem ser fracionárias, essa solução será tão boa ou melhor que a solução inteira e;
- Como a função objetivo multiplica e soma valores inteiros, esse resultado poderia ser um inteiro.

Assim o método de Gomory é uma adaptação do modelo linear, e a dificuldade encontrada na resolução deste tipo de problema se dá pela falta de consistência na resolução de problemas com inteiros. Apesar de saber-se que um número finito de iterações chegaria à convergência do problema, a implementação computacional (com seus erros de arredondamento) se torna muitas vezes desafiadora (TAHA, 2008).

De acordo com Hillier e Lieberman (2013), problemas de PI tem um número bem menor de soluções a serem consideradas do que um problema de PL. Problemas de PI que contam com uma região de soluções limitadas, têm um número finito de soluções, porém esse número finito de soluções pode ser muito grande. Por exemplo, para um problema com n variáveis, há 2^n soluções consideráveis, portanto a cada uma variável que é adicionada a n , o número de soluções dobra, ou seja há um crescimento exponencial na dificuldade do problema.

Neste trabalho, serão utilizados os conceitos de programação linear inteira (PLI) para modelar o problema de otimização do sequenciamento das tarefas nas máquinas em ambiente paralelo com tempo de processamento das tarefas diferente por máquina.

2.2 PLANEJAMENTO, PROGRAMAÇÃO E CONTROLE DE PRODUÇÃO

A produção de bens de consumo de maneira automatizada, teve início com a Revolução Industrial, que foi quando se tornou possível a produção em massa. Com isso os sistemas de planejamento e controle da produção evoluíram também, e como consequência, houve a evolução da ciência da administração (LUSTOSA *et al.*, 2008).

A diferença básica entre os termos planejamento e programação está no horizonte de tempo, enquanto o planejamento está ligado a um maior horizonte de tempo (médio e longo prazos), a programação preocupa-se com o curto prazo. As quantidades a serem produzidas, materiais e componentes necessários para esta produção, bem como os recursos necessários, como mão de obra, para produção ao longo do período planejado são determinados nesta atividade.

O planejamento de produção pode ser visto como um fator chave nas decisões da empresa por ser a ligação entre a gestão da produção e inventários, com as decisões de planejamento e outras áreas funcionais. Trata-se da integração da manufatura com plano de vendas e setor financeiro através do planejamento integrado (FAVARETTO, 2001).

Segundo Chiavenato (1990) a finalidade do planejamento da produção é obter simultaneamente a melhor eficiência e eficácia do processo produtivo. Ou seja, o planejamento da produção procura definir antecipadamente o que se deve fazer, quando fazer, quem deve fazer e como fazer.

O planejamento da produção é o estabelecimento *a priori* daquilo que a empresa deverá produzir, tendo em vista, de um lado, a sua capacidade de produção e, de outro lado, a previsão de vendas que deve ser atendida. O PPCP envolve um conjunto de funções integradas que visam orientar o processo produtivo em função dos objetivos da empresa e dos recursos empresariais disponíveis (CHIAVENATO, 1990).

A programação da produção trata da determinação antecipada da produção, a médio prazo, dos produtos a serem produzidos por uma empresa. Representa o que deve ser produzido em termos de quantidades e de datas, tanto de início da produção quanto de fim. Leva em consideração a previsão de demanda, a carteira de pedidos, disponibilidade dos recursos necessários e a capacidade, a fim de estabelecer a melhor estratégia de produção (RUSSOMANO, 1995).

Determina quantidades a serem produzidas, levando em consideração a disponibilidade dos recursos que serão utilizados para a produção ao longo de período pré-determinado, que pode abranger semanas ou dias. Para Porter (1999), para que

uma empresa encontre uma boa solução para seu planejamento, programação e controle de produção (PPCP), é necessário que seus principais processos de negócio sejam entendidos.

O controle da produção (CHIAVENATO, 2008) trata-se de uma das funções administrativas mais importantes, pois é encarregada por medir e corrigir o desempenho da produção. A tarefa do controle da produção é a de verificar se o setor “produção” está seguindo as diretrizes do planejamento. É um processo cíclico e repetitivo, e por ter esse perfil a tendência é que os desvios sejam cada vez menores, e que com o tempo consiga-se aperfeiçoar o processo produtivo.

Faz parte do controle da produção atividades relacionadas à coleta dos dados reais do processo produtivo, com a finalidade de determinar possíveis desvios no processo, possibilitando assim ações preventivas ou até mesmo corretivas. Com essas informações pode-se identificar problemas na produção, como problemas em máquinas, com a ordem de produção, sendo possível identificar tendências passíveis de correção.

Lustosa *et al.* (2008) aponta que as decisões a respeito da programação de produção dividem-se em categorias:

- Designação: que aponta onde a produção da tarefa será executada;
- Sequenciamento: que determina a sequência com que as tarefas devem ser produzidas em um dado recurso;
- Programação: responsabiliza-se por determinar quando uma tarefa deverá ser iniciada e quando deverá ser terminada, não apenas a sequência;
- Despachos: determina quando e para quem a ordem será emitida;
- Controle: acompanhamento dos trabalhos a fim de garantir que a programação seja seguida e apontamento de eventuais necessidades de reprogramação ou intervenção;
- Apressamento: acelerar uma ordem de produção pelo aumento de sua prioridade a fim de atender a uma necessidade específica;
- Carregamento de oficinas: define os roteiros e programa as tarefas.

Para Lustosa *et al.* (2008) é de responsabilidade do PPCP a coordenação e aplicação dos recursos produtivos de modo a atender os planos estabelecidos nos níveis estratégicos, táticos e operacional da melhor maneira possível. A Figura 2 ilustra a pirâmide da competitividade empresarial, apresentada pelo autor.

Figura 2 – PPCP: base para a competitividade empresarial



Fonte: Adaptado de Lustosa *et al.* (2008)

O sistema de PPCP se ocupa do planejamento e controle de todos os aspectos da produção, inclusive de gerenciamento de materiais, da programação das máquinas e pessoas e da coordenação de fornecedores e clientes-chave, garantindo um bom relacionamento entre os setores (VOLLMAN *et al.*, 2006).

2.2.1 Objetivos do PPCP

Segundo Favaretto (2001), "o objetivo principal do PPCP é comandar o processo produtivo, transformando informações de vários setores em ordens de produção".

De acordo com Chiavenato (2008), os objetivos da programação da produção incluem:

- Coordenar e integrar todos os envolvidos no processo produtivo da empresa, sejam eles diretos ou indiretos;
- Garantir que as entregas dos produtos aos clientes sejam feitas na data correta (requerida ou prometida);
- Garantir a disponibilidade de matéria prima e de componentes para a produção;
- Balancear o processo produtivo para que não haja gargalos ou ociosidades;
- Coordenar os recursos, como compras, mão de obra, máquinas e equipamentos de maneira racional;
- Distribuir o trabalho de forma proporcional, a fim de assegurar a melhor sequência da produção e o melhor resultado em termos de eficiência e eficácia.

Em suma a função do PPCP é a organização, padronização e sistematização do processo produtivo, levando a empresa a produzir de uma maneira mais segura, rápida, eficiente e com o menor custo possível.

Existem três níveis hierárquicos de planejamento e controle, o nível estratégico, de longo prazo, o nível tático, de médio prazo e o nível operacional responsável pelo curto prazo (BARROS *et al.*, 1998).

Outro objetivo do PPCP é conduzir da maneira mais eficiente e eficaz possível as decisões para cada um dos níveis hierárquicos: estratégico, tático e operacional. É o setor responsável por garantir a continuidade da produção em uma indústria, coordenar e administrar os recursos que estão disponíveis e que serão utilizados durante o processo produtivo. (SLACK *et al.*, 2002), definir prazos através do sequenciamento das ordens de produção, alocando os recursos durante um tempo para a realização das tarefas. Porém o problema pode ter inúmeras formas e restrições (ARAUJO,2013).

2.2.2 Etapas do PPCP

De acordo com Erdmann (1998), pode se compreender as funções de programação e controle em etapas, que se sintetizam como segue:

- Definição da necessidade de produtos finais: a programação é iniciada com a quantificação de cada produto a ser produzido e também quando os mesmos deverão estar disponíveis; esses dados consideram as vendas ou a previsão delas, capacidade produtiva da fábrica e política administrativa;
- Cálculo das necessidades de material: trata-se do cálculo dos componentes, peças e matérias primas necessárias para produzir os produtos finais. E se na data para disponibilidade dos produtos, haverá o material necessário;
- Definições de prazos, capacidades e ajustes: o prazo de entrega do produto final exige prazos intermediários e conseqüentemente necessidade de capacidades específicas. Eventuais divergências requerem ajustes do processo ou do prazo junto ao cliente;
- Liberação da produção: feitos os procedimentos anteriores, pode-se então determinar o início da produção, e isto se faz mediante critérios de liberação (ordenamento das tarefas);
- Controle: é um acompanhamento, em que se verifica o que é produzido e se compara com o que foi programado.

O mesmo autor menciona que a programação é responsável por garantir o cumprimento dos prazos estabelecidos, datas e fim das operações, garantir os insumos necessários para a produção, alocar os recursos, prever e contornar os pontos críticos na produção, prever capacidade ociosa e otimizar o sequenciamento da produção.

Para Moreira (1986), controle é responsável por dominar, governar e fiscalizar o processo produtivo. A prioridade é de controlar a produção para que não haja desvio das normas estabelecidas, através de fases que compreendem estabelecimento de padrões, avaliação de desempenho, comparação do desempenho com o padrão estabelecido e ação corretiva.

2.3 SEQUENCIAMENTO DA PRODUÇÃO

Segundo Le Pape (2005) um problema de escalonamento consiste em decidir quando cada atividade deve ser executada, levando em conta as restrições de tempo e recurso.

Para Hochbaum (1999) os objetivos em problemas de escalonamento são:

- Minimizar o tempo de execução total, medindo o nível de utilização das máquinas;
- Minimizar o tempo de espera de cada tarefa, que se trata do tempo entre a finalização de uma tarefa até o início da próxima;
- Minimizar os custos de execução das atividades.

O sequenciamento, corresponde à ordenação entre as diversas tarefas de um processo ou à ordem pela qual as tarefas serão processadas em uma determinada máquina (PINEDO, 2008).

Ainda em Pinedo (2008) nos problemas de escalonamento, tanto o número de tarefas quanto o número de máquinas são finitos. O número de tarefas é denotado por n e o número de máquinas por m .

O autor estabelece ainda que trata-se de um processo que visa definir a melhor sequência de processamento das tarefas, gerenciando a escassez de recursos ao longo do tempo. Portanto, é um processo de tomada de decisão que visa otimizar os objetivos estabelecidos.

A programação, para Pinedo (2008) atua na alocação de recursos para realização de tarefas em um determinado período de tempo, trata-se de um processo que envolve decisão com a meta de otimizar um ou mais objetivos. Esses objetivos podem ser diversos, como a redução do tempo total de processamento de um conjunto de tarefas, ou minimizar as tarefas completadas com atraso.

É função do sequenciamento a busca pela adequação dos limitados recursos de produção, sejam eles, mão de obra, máquinas, instalações, materiais, entre outros. Isso dependerá do grau de detalhamento e intensidade que se dá a execução das

funções de sequenciamento no sistema produtivo em que se encontra inserido (TUBINO, 2007).

Segundo Arnold (1998), *Scheduling* é o processo definido por estabelecer as datas de começo e fim para as operações necessárias para a produção de um item. Para isso o planejador necessita de informações sobre o roteiro, capacidade produtiva, e tempo de processamento dos centros de trabalho envolvidos.

Segundo Graves (1981), em *Scheduling* trabalha-se com operação no tempo de processamento, que é formado por cinco elementos:

- Tempo em fila, é o tempo em que um trabalho aguarda para ser iniciado em um centro de produção;
- Tempo de *setup*, que é o tempo necessário para preparar um centro de trabalho para executar a operação;
- Tempo de corrida, é o tempo em que a operação é realizada;
- Tempo de espera, tempo de aguardo pelo material para ser transferido ao próximo centro de trabalho;
- Tempo de movimentação, tempo de transito entre os centros de trabalho.

Os objetivos do sequenciamento normalmente estão associados a regras de priorizações, Tubino (2007) estabelece que essas regras pertencem a um procedimento simplificador que levam em consideração informações sobre tarefas e sobre as condições do sistema produtivo. Visa-se selecionar as tarefas que terão prioridade de processamento, alocando assim os recursos de maneira a executar essas tarefas.

2.3.1 Sequenciamento de Tarefas

Para Joo e Kim (2015), o sequenciamento de tarefas é uma etapa muito importante no processo de fabricação, por se tratar de uma série de atividades que tornam disponíveis os recursos necessários para a conclusão das tarefas, determinando horários e datas para a produção. A complexidade de um sistema de sequenciamento de produção se dá pela dinâmica dos processos produtivos, pela

chegada de pedidos inesperados, flexibilidade ou alteração nas prioridades de entrega e pelas taxas de defeitos aleatórias.

Segundo Pinedo (2008) sequenciamento é um processo de tomada de decisões que visa definir a melhor sequência de processamento de tarefas, sujeito a restrições de recursos ao longo do tempo, com objetivo de otimização que incluem:

- *Makespan*: tempo total da programação de tarefas;
- *Mean flow-time*: média de tempo de duração do fluxo;
- *Total flow-time*: tempo total de duração do fluxo;
- *Mean tardiness*: atraso médio do fluxo de tarefas;
- *Maximum tardiness*: atraso máximo para a conclusão de tarefas;
- *Tardiness*: soma das penalidades por atraso;
- *Earliness*: soma das penalidades por adiantamento.

Decisões acerca da ordenação das tarefas no sistema devem ser tomadas a partir do momento em que as mesmas tornam-se aptas ao processamento. Tais decisões podem se tornar complexas, gerando lentidão na tomada de decisão (SLACK *et al.*, 2002).

Os objetivos do sequenciamento para Tubino (2007) estão normalmente associados a regras de priorização que pertencem a um procedimento simplificador, que consideram informações sobre as tarefas e condições do sistema produtivo, visando selecionar as tarefas, e os recursos necessários para execução dessas tarefas, que terão prioridade de processamento.

2.3.2 Sequenciamento em Máquinas

Sequenciamento de máquinas são as diretrizes sobre a ordem em que a produção será realizada, respeitando prioridades e as restrições impostas pelo processo industrial. Para classificar os modelos de sequenciamento necessita-se caracterizar os dados sobre os recursos e o comportamento das operações. Pode haver um ou vários tipos de recursos, que podem ser especializados (um recurso

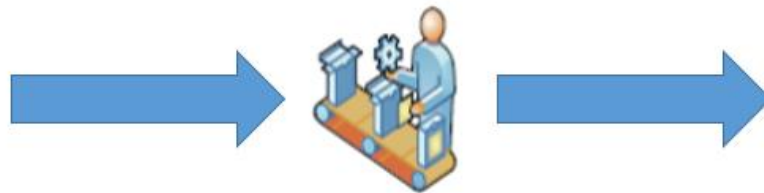
executa um determinado tipo de operação) ou paralelos (diversos recursos executando uma mesma operação) (BAKER, 1974).

De acordo com Pinedo (2008) o sequenciamento de máquinas pode ser classificado quanto ao modelo de máquina utilizado ou quanto ao ambiente de produção. Os possíveis ambientes de máquinas considerados por esse autor são descritos a seguir.

2.3.2.1 Modelo de máquina única

Para o autor o modelo de máquina única trata-se do modelo mais simples quando comparado com os demais possíveis, neste modelo todos os produtos são processados em uma única máquina especializada. Considera-se a existência de apenas um recurso, podendo ser de fato um único equipamento, ou até mesmo uma célula de produção ou um conjunto de recursos sendo modelados como um único. Na Figura 3 é apresentado um esquema deste tipo de problema.

Figura 3 – Exemplo de processamento em máquina única



Fonte: Santo (2013)

2.3.2.2 Modelo de máquina em paralelo

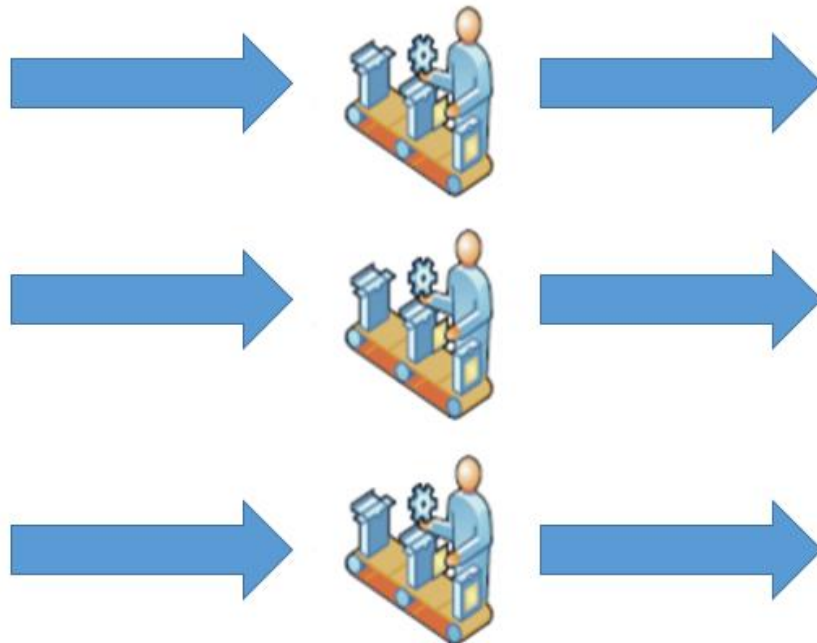
Ainda segundo o autor o modelo de máquinas paralelas pode considerar máquinas idênticas ou com diferença no tempo de processamento, em paralelo, ou ainda máquinas não relacionadas em paralelo. No modelo de máquinas idênticas em paralelo, existem m máquinas e os produtos podem ser produzidos em qualquer uma das máquinas ou em qualquer uma de um subconjunto.

No modelo de máquinas com diferença de velocidade de processamento em paralelo, existem m máquinas em paralelo com diferença na velocidade. A velocidade da máquina i é denotada por v_i , o processamento é definido por p_j e o tempo de processamento do produto j na máquina i é dado por $\frac{p_j}{v_i}$.

O modelo de máquinas não relacionadas em paralelo é um modelo generalizado em relação aos anteriores, onde existem m máquinas diferentes em paralelo. O produto j pode ser processado com velocidade v_{ij} na máquina i .

Trata-se de um conjunto de máquinas que realizam a mesma operação, podendo ser idênticas ou não. Se idênticas, possuem a mesma velocidade de produção, se não idênticas, possuem velocidades diferentes para a mesma operação ou podem ainda ser máquinas totalmente diferentes. Na Figura 4 é apresentado um exemplo desse modelo. Para o estudo de caso tratado neste trabalho, por haver um certo número finito de máquinas que executam uma gama de tarefas com tempos de processamento diferentes, será considerado o modelo de máquinas em paralelos com diferentes tempos de processamento.

Figura 4 - Exemplo de processamento em máquinas paralelas



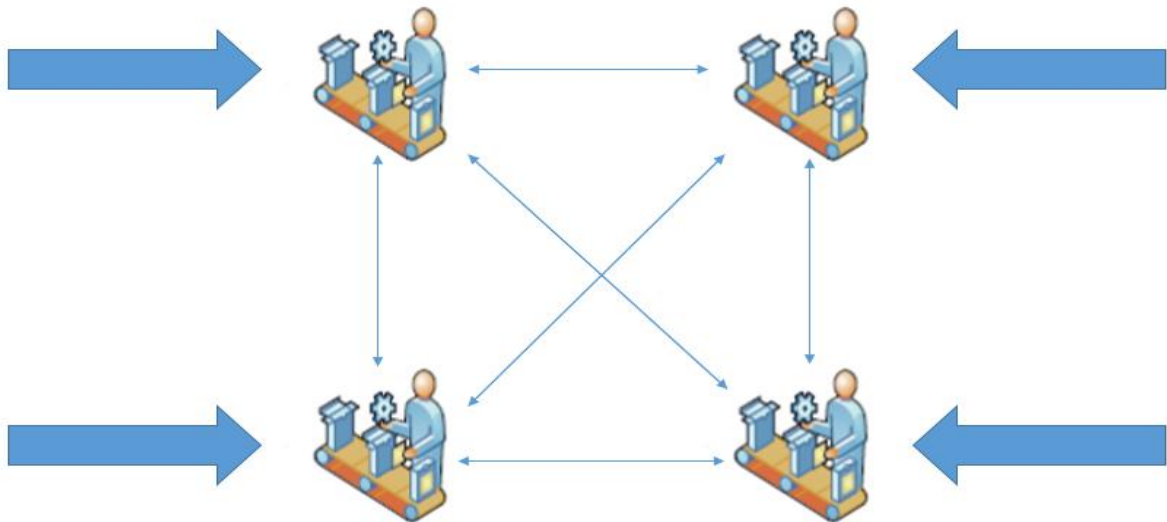
Fonte: Santo (2013)

2.3.2.3 Modelo de Job Shop

Segundo o autor, no modelo de *Job Shop*, existem m máquinas e cada produto tem um roteiro pré-determinado. Existe uma distinção entre produções onde cada produto deve visitar uma sequência de máquinas uma única vez e produções onde as máquinas podem ser visitadas mais de uma vez.

Neste modelo, cada máquina é modelada como um centro de trabalho e cada operação possui um roteiro próprio de fabricação, onde os produtos não precisam necessariamente passar por todas as máquinas, nem seguir a mesma sequência de produção. Consiste em um processamento de n peças em m máquinas, onde o processamento de cada peça consiste em x operações realizadas em uma sequência específica, como na Figura 5.

Figura 5 - Exemplo de processamento em máquinas em *Job Shop*



Fonte: Santo (2013)

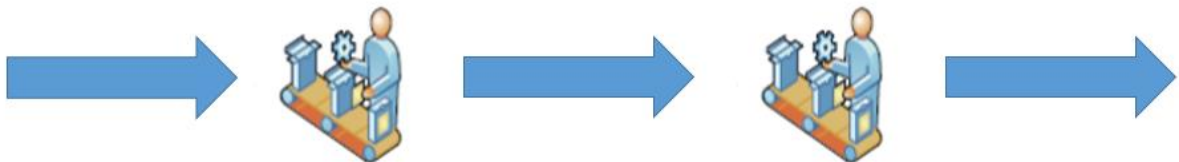
2.3.2.4 Modelo flow shop

De acordo com o autor no modelo *flow shop* existem m máquinas em série. Onde cada produto deve ser processado em cada uma das m máquinas em uma mesma sequência predeterminada.

Depois de completar o processamento em uma máquina o item deve entrar na fila para ser processado na próxima máquina, fila essa que geralmente opera no esquema *first in first out* (FIFO).

Trata-se de um caso particular de *Job Shop*, onde todos os *jobs* possuem uma mesma sequência tecnológica de produção, várias operações em série. Cada operação deve ser processada em todas as máquinas da linha em uma ordem pré-estabelecida, como na Figura 6.

Figura 6 - Exemplo de processamento em máquinas em *Flow Shop*



Fonte: Santo (2013)

2.3.2.5 Modelo de Flexible Flow Shop

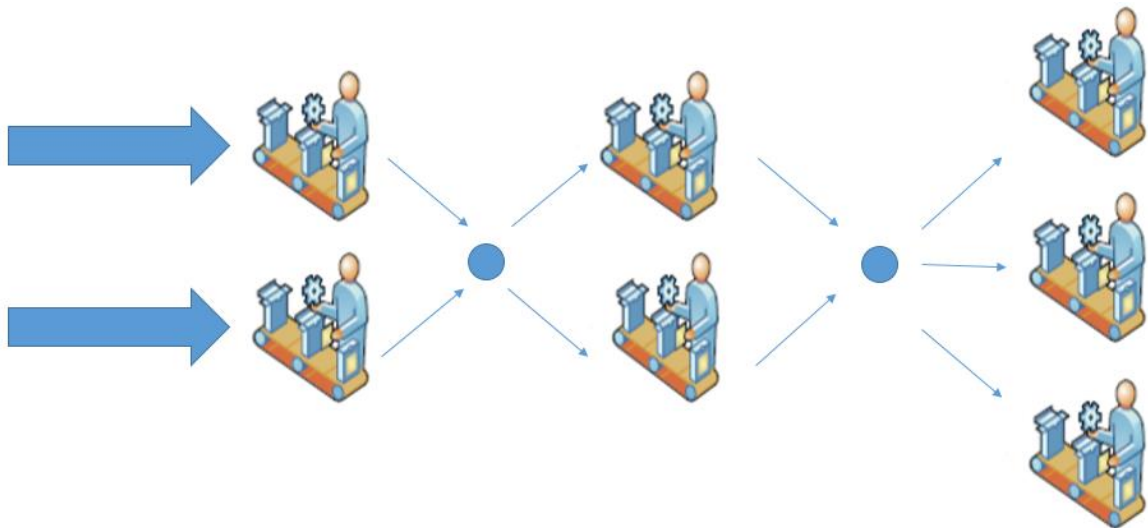
Ainda de acordo com o autor, o modelo *flexible flow shop* é uma generalização do modelo *Flow Shop* e dos ambientes de máquinas em paralelo, em que ao invés de m máquinas em série, existem c etapas em série e em cada etapa há máquinas idênticas em paralelo.

Cada produto deve ser processado no estágio 1, depois no estágio 2 e assim por diante. Um estágio funciona como um banco de máquinas em paralelo.

As filas entre os estágios não precisam necessariamente serem processadas de acordo com o esquema de *first in first out*. A Figura 7 apresenta um exemplo desse modelo.

Os círculos na figura representam os pontos intermediários entre as etapas do modelo.

Figura 7 - Exemplo de processamento em máquinas em *Flexible Flow Shop*



Fonte: Santo (2013)

Difere do modelo anterior apenas pelo fato de que cada estágio da linha de produção pode ser composto por máquinas em paralelo, permitindo-se assim uma maior flexibilidade da produção.

2.4 MÉTODOS DE SOLUÇÃO

Os problemas de programação linear, dependendo das suas características, trazem complexidades variadas e por isso, podem ser resolvidos considerando-se diversas abordagens, as quais podem trazer soluções exatas ou aproximadas.

2.4.1 Métodos Heurísticos

Para Hillier (2013) o método heurístico é um procedimento que procura uma solução viável excelente, mas não necessariamente encontrara a solução ótima para o problema. Com as meta-heurísticas, que são heurísticas melhoradas, criou-se estratégias de alto nível capaz de escapar dos ótimos locais e realizar uma busca consistente de soluções viáveis.

As meta-heurísticas incluem a programação de lógica de restrição; busca aleatória gulosa; computação evolutiva natural; métodos de pesquisa espacial; busca tabu; redes neurais; estratégias de pesquisa não monotônicas; entre outras (OSMAN; LAPORTE, 1996).

São exemplos de heurísticas e meta-heurísticas: Vizinho mais próximo; Algoritmos Genéticos; Simulated Annealing; Busca Tabu; Colônia de formigas. Diversos autores utilizaram das heurísticas e meta-heurísticas para a resolução de problemas de sequenciamento de tarefas em máquinas, alguns deles são apresentados a seguir.

Gomes Júnior (2007) propôs uso de métodos heurísticos baseados nas meta-heurísticas *Greedy Randomized Adaptive Search Procedure* (GRASP) e *Iterated Local Search* (ILS) para resolução do problema de sequenciamento em uma máquina com penalidade por antecipação e por atraso da produção, o autor considerou tempo de preparação da máquina dependente da sequência de produção e janela de entrega dos produtos. Com base na sequência de *Jobs* gerados utilizou um algoritmo para determinar a data ótima de início do processamento. Para problemas de pequenas dimensões (8 a 12 *jobs*) alcançou-se o valor ótimo, e em problemas de dimensões maiores (15 a 75 *jobs*) obteve-se desvios baixos em relação à melhor solução obtida.

Já Köhler (2004) propõe métodos comparados com as heurísticas Busca Tabu e Algoritmos Meméticos para resolver o problema de sequenciamento de n tarefas em m máquinas paralelas idênticas, com o objetivo de minimizar o tempo máximo de finalização (*makespan*), considerando o tempo de preparação dependente da sequência. Para instâncias de pequeno porte, comparações são feitas com o valor

ótimo obtido através de uma busca dicotômica. Para instâncias maiores, as comparações demonstram a robustez e boa qualidade das soluções.

Monaci *et al.* (2015) abordou o problema de sequenciamento de tarefas em máquinas paralelas não relacionadas com o objetivo de minimizar o tempo máximo de conclusão do sequenciamento (*makespan*), considerando restrições de precedência entre as tarefas e tempo de preparação dependentes da sequência. A autora propõe uma heurística construtiva que utiliza uma regra de prioridade para selecionar uma tarefa a ser alocada e sequenciada em uma determinada máquina.

Uma outra abordagem pode ser encontrada em Santos (2016), em que o objetivo é encontrar a otimização do sequenciamento de tarefas em cada máquina de tal maneira que os desgastes das máquinas sejam reduzidos, minimizando assim o tempo de conclusão de todas as tarefas. A autora utiliza algoritmos baseados nas meta-heurísticas *Iterated Local Search* (ILS) e *Iterated Greedy* (IG). Os algoritmos são comparados entre si e também com um algoritmo *Simulated Annealing* (AS).

Fidelis e Arroyo (2017) sugere o uso de duas heurísticas *Adaptative Large Neighborhood Search* (ALNS) e *Simulated Annealing* (AS) com avaliação de desempenho através de experimentos computacionais comparando com duas heurísticas da literatura *Memetic Algorithm* e *Variable Neighborhood Search* para a resolução de problemas de sequenciamento de tarefas em máquinas paralelas idênticas de processamento em lotes. Um lote consiste em um grupo de tarefas que podem ser processadas simultaneamente na mesma máquina, a fim de determinar os lotes de tarefas a sequenciar nas máquinas de maneira a minimizar o atraso total ponderado.

Noal (2015) aborda o problema de programação de tarefas em sistemas de pintura com tempos de *setup* dependentes da sequência do processamento, propondo uma sistemática para liberação do tamanho de lote que minimize a soma dos custos de *setup* e pintura. Os lotes liberados são, então, sequenciados utilizando as regras *Apparent Tardiness Cost* (ATC) e *Apparent Tardiness Cost with Setups* (ATCS).

Outro uso de heurística é abordado por Müller e Limberger (2000). Os autores resolvem o sequenciamento de tarefas independentes de forma não preemptiva em sistemas de processadores uniformes, com o objetivo de minimizar o tempo de

execução (*makespan*), considerando um conjunto de n tarefas, onde as tarefas possuem um tempo de processamento. Para tal desenvolveram uma heurística de trocas composta de três fases: alocação inicial, balanceamento de carga e fase de dupla troca.

Destaca-se que na literatura tem-se bastante problemas de sequenciamento resolvido com uso de métodos heurísticos, mas a principal contribuição do trabalho está na resolução pelo método exato, que será explorado a seguir.

2.4.2 Método exato

Dentre os diversos métodos exatos de resolução de modelos matemáticos, destacam-se o método *Simplex* e o método *Branch-and-Bound*, para problemas lineares e inteiros, respectivamente.

O método *Simplex* desenvolvido por George Dantzig em 1947, provou-se ser eficiente e que pode ser utilizado para solucionar problemas de grande porte nos computadores atuais (HILLIER, 2013).

Segundo Collin (2013), o método caminha de uma solução viável para outra, até que o valor da função objetivo alcance o valor ótimo. O método se divide em três partes que podem ser definidas como: início (preparação dos dados de entrada), iteração (repetição do procedimento de otimização) e critério de parada (para avaliar se o modelo tem solução ótima, ou, se é impossível obtê-la).

O algoritmo *Branch-and-Bound* foi desenvolvido por A. Land e G. Doing em 1960 (TAHA, 2008) baseia-se em desenvolver uma enumeração dos pontos candidatos à solução ótima. O termo *Branch* faz referência ao fato que realiza-se partições no espaço das soluções e o termo *Bound* ressalta a otimalidade da solução (GOLDBARG, 2005). Utiliza uma estrutura dinâmica de árvore em que nós folha representam as várias soluções possíveis, a busca começa em um nó raiz que é expandido aumentando a amplitude de busca.

Segundo Collin (2013), o algoritmo começa com a solução ótima, desconsiderando restrições de variáveis inteiras. A cada iteração, ele subdivide o problema original e ao mesmo tempo elimina partes fracionárias da solução não inteira, adicionalmente, em cada iteração, elimina partes inteiras do problema subdividido.

Devido às restrições de tempo computacional existente nos métodos exatos não há muitos trabalhos na literatura a respeito da resolução do problema de sequenciamento de tarefas em máquinas, havendo apenas para casos pequenos com poucas máquinas e poucas tarefas.

Johnson (1954) desenvolveu um algoritmo exato para resolver o problema para duas máquinas que processam dois produtos (2x2), cuja ideia central é iniciar a execução da tarefa com menor tempo de processamento na primeira máquina, permitindo que seu processamento na segunda máquina comece o mais cedo possível e terminar executando a tarefa com menor tempo de processamento na segunda máquina minimizando o tempo em que a primeira fica ociosa.

Kuo e Yang (2005), resolveram o problema de sequenciamento de tarefas em máquina única com tempo de processamento dependente da sequência. Os autores apresentam uma solução que permanece com tempo polinomialmente solucionável para o objetivo que é de minimizar o tempo de conclusão das tarefas e apresenta a sequência obtida.

Yang e Chern (1995) consideraram um problema de sequenciamento com duas máquinas em ambiente *flow shop* com restrição de tempo de espera, isso significa que, para cada trabalho, o tempo entre as máquinas não pode ser maior que um determinado limite superior. O objetivo é minimizar o *makespan*, que é o tempo total que o produto leva para ser finalizado. Os autores propõem o uso do algoritmo de *Branch-and-bound*.

Com avanços computacionais e desenvolvimento de máquinas de alta performance, tornou-se possível resolver problemas reais com número grande de variáveis, que antes não era possível utilizando métodos exatos. Neste contexto, surgiu o site NEOS Server, um projeto entre algumas universidades que utilizam solver comerciais em máquinas de alta performance, trata-se de um serviço gratuito para

solução de problemas de otimização sem um limite de variáveis. Os computadores executam o modelo remotamente via *website* (PALOMARES, 2016).

3 MODELO MATEMÁTICO PARA SEQUENCIAMENTO EM MÁQUINA

Inicialmente, neste capítulo, será apresentada a caracterização do problema a ser considerado neste trabalho. Na sequência serão detalhados os procedimentos para a generalização do modelo (2.16) a (2.27) para o sequenciamento de tarefas em várias máquinas em paralelo, e posteriormente, apresentado o método de solução.

3.1 MÁQUINA ÚNICA

O exemplo a seguir considera uma única máquina para processar três tarefas com tempos de processamento e prazos de execução diferentes (TAHA, 2008). Esse exemplo foi utilizado de base para o problema expandido. Quando não for cumprido o prazo de entrega incide-se uma multa proporcionalmente ao tempo de atraso.

A Tabela 1 apresenta os tempos de processamento, prazos de execução e os valores da multa por atraso.

Tabela 1 – Dados para sequenciamento em máquina única

Tarefa	Processamento (p_i em dias)	Prazo de execução (d_i em dias)	Multa por atraso \$/dia
1	5	25	19
2	20	22	12
3	15	35	34

Fonte: Taha (2008)

O objetivo do problema é determinar a sequência de processamento das três tarefas que resulte na multa mínima por atraso.

3.1.1 Restrições para o sequenciamento em uma máquina

Para apresentar um modelo de otimização matemática que determine tal sequência ótima, definiu-se:

$$x_j = \text{data de início da tarefa } j, \text{ a partir do dia } 0 \quad (2.7)$$

Deve-se considerar dois tipos de restrições para o problema:

- I. Restrições de não interferência entre as tarefas, ou seja, restrições que garantam que duas tarefas não sejam executadas ao mesmo tempo;
- II. Restrições de prazo de execução.

As restrições de não interferência garantem que duas tarefas i e j , com tempo de processamento p_i e p_j não serão processadas simultaneamente se o tempo de início da tarefa i é maior ou igual ao tempo de início da tarefa j acrescido do tempo de processamento da tarefa j , que matematicamente pode ser expresso por:

$$x_i \geq x_j + p_j, \text{ se } i \text{ preceder } j, \quad (2.8)$$

ou, se o tempo de início da tarefa j é maior ou igual ao tempo de início da tarefa i acrescido do tempo de processamento da tarefa i , ou seja, se:

$$x_j \geq x_i + p_i. \quad (2.9)$$

Como todos os modelos de otimização matemática consideram somente restrições simultâneas, é necessário transformar as restrições ou-ou em restrições simultâneas (e-e). Para isso, introduz-se a variável binária auxiliar:

$$y_{ij} = \begin{cases} 1, & \text{se } i \text{ preceder } j \\ 0, & \text{se } j \text{ preceder } i \end{cases} \quad (2.10)$$

Então, considerando-se uma constante MM positiva grande, a restrição ou-ou (2.8) e (2.9) são convertidas para as seguintes restrições simultâneas:

$$MMy_{ij} + (x_i - x_j) \geq p_j \quad (2.11)$$

e

$$MM(1 - y_{ij}) + (x_j - x_i) \geq p_i. \quad (2.12)$$

Essa conversão garante que apenas uma das duas restrições (2.11) e (2.12) possa ser ativada.

Quando $y_{ij} = 0$, a restrição (2.11) é ativada e a restrição (2.12) é redundante, pois no seu lado esquerdo será incluído MM , que é muito maior do que p_i .

Da mesma forma, se $y_{ij} = 1$, a restrição (2.11) é redundante e a restrição (2.12) é ativada.

Para garantir a segunda restrição, ou seja, a restrição de prazo de execução, considerando s_j uma variável irrestrita, o prazo de execução da tarefa j pode ser dado por:

$$d_j = x_j + p_j + s_j. \quad (2.13)$$

Quando $s_j \geq 0$, o prazo é cumprido, e se $s_j < 0$, a multa é aplicada pelo atraso.

Considerando a substituição:

$$s_j = s_j^- - s_j^+, \quad (2.14)$$

com $s_j^-, s_j^+ \geq 0$, então a restrição (2.13) se torna:

$$x_j + s_j^- - s_j^+ = d_j - p_j. \quad (2.15)$$

Com isso, o custo da multa por atraso é proporcional a s_j^+ .

O modelo completo para o problema é então dado pelas restrições (2.16) a (2.27). Observa-se que este modelo para determinar o sequenciamento de três tarefas

em uma única máquina possui doze variáveis, sendo três não negativas e as demais variáveis inteiras, e possui nove restrições.

(2.16)

$$\text{Minimizar } Z = 19s_1^+ + 12s_2^+ + 34s_3^+$$

S:a:

$$x_1 - x_2 + My_{12} \geq 20 \quad (2.17)$$

$$-x_1 + x_2 - My_{12} \geq 5 - M \quad (2.18)$$

$$x_1 - x_3 + My_{13} \geq 15 \quad (2.19)$$

$$-x_1 + x_3 - My_{13} \geq 5 - M \quad (2.20)$$

$$x_2 - x_3 + My_{23} \geq 15 \quad (2.21)$$

$$-x_2 + x_3 - My_{23} \geq 20 - M \quad (2.22)$$

$$x_1 + s_1^- - s_1^+ = 25 - 5 \quad (2.23)$$

$$x_2 + s_2^- - s_2^+ = 22 - 20 \quad (2.24)$$

$$x_3 + s_3^- - s_3^+ = 35 - 15 \quad (2.25)$$

$$x_1, x_2, x_3, s_1^-, s_1^+, s_2^-, s_2^+, s_3^-, s_3^+ \geq 0 \quad (2.26)$$

$$y_{12}, y_{13}, y_{23} = (0,1) \quad (2.27)$$

Considerando-se a resolução com o Solver do Excel, ou o Lingo, e como pode ser observado em (TAHA, 2008), adotando-se $MM = 100$, a solução ótima para esse problema é dada pela sequência $2 \rightarrow 1 \rightarrow 3$, com $x_1 = 20$, $x_2 = 0$ e $x_3 = 25$, ou seja, a tarefa 2 tem início no tempo 0 e é concluída em $0 + 20 = 20$ dias, a tarefa 1 começa no tempo 20 com conclusão em $20 + 5 = 25$ dias, e a última tarefa inicia-se no tempo 25 e termina em $25 + 15 = 40$ dias.

Observe que a tarefa 3 terá um atraso de $40 - 35 = 5$ dias em relação ao prazo de entrega, gerando uma multa de $5 \times \$34 = \170 .

3.1.2 Modelo em AMPL para o sequenciamento em uma máquina

O modelo em AMPL para este exemplo pode ser visto na Figura 8.

Figura 8 – Modelo em AMPL

```

----- Arquivo modelo em AMPL -----
param n;
param MM;
param p {i in 1..n};
param d {i in 1..n};
param Rate {i in 1..n};
var x{i in 1..n} ≥ 0;
var y{i in 1..n, j in 1..n} binary;
var sM{i in 1..n} ≥ 0;
var sP{i in 1..n} ≥ 0;
minimize PerDayPenalty: sum{i in 1..n} Rate[i] * sP[i];
subject to C1 {j in 1..n, i in 1..n : i ≠ j}: MM * y[i, j] + (x[i] - x[j]) >= p[j];
subject to C2 {j in 1..n, i in 1..n : i ≠ j}: MM * (1 - y[i, j]) + (x[j] - x[i]) >= p[i];
subject to C3 {j in 1..n}: x[j] + sMM[j] - sP[j] = d[j] - p[j];

```

Fonte: Baseado em (TAHA, 2008)

O arquivo com os dados do problema pode ser apresentado como na Figura 9.

Figura 9 – Arquivo para entrada de dados

```

----- Arquivo de dados -----
data;
param n = 3;
param MM = 100;
param p: 1 2 3 := 5 20 15;
param d: 1 2 3 := 25 22 35;
param Rate: 1 2 3 := 19 12 34;

```

Fonte: Baseado em (TAHA, 2008)

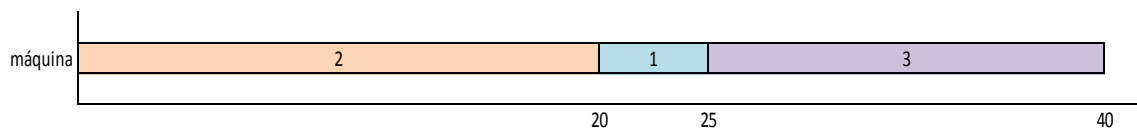
O modelo em AMPL da Figura 8 pode ser resolvido pelo Solver CPLEX ou GUROBI disponível no site NEOS Server (NEOS GUIDE SERVER, 2017).

Os mesmos valores encontrados na resolução com o Solver do Excel e apresentados no tópico 2.5.2 foram encontrados com o uso do Solver GUROBI.

Encontrou-se a resposta após vinte e sete iterações. São quinze variáveis e onze restrições, das quais seis variáveis são binárias, e nove variáveis lineares e três restrições de igualdade e doze restrições de desigualdade. A função objetivo como sabe-se, é linear.

Na figura 10 é apresentado o sequenciamento obtido.

Figura 10 – Sequência de tarefas em uma máquina



Fonte: Autoria própria

O modelo apresentado nesta seção pode ser visto em (TAHA, 2008) e servirá de base para a construção do modelo considerando-se várias máquinas.

3.2 SEQUENCIAMENTO DE TAREFAS EM VÁRIAS MÁQUINAS

O problema abordado neste estudo considera dados reais de uma empresa multinacional do setor plástico que além das fábricas localizadas no Brasil, conta com fábricas em outros países. Os dados utilizados neste trabalho referem-se a fábrica que se encontra nos Estados Unidos da América.

Dada à utilização de informações e procedimentos operacionais sigilosos a identidade da empresa será preservada e os dados apresentados podem estar mascarados.

A fábrica em questão contém um total de 24 máquinas de injeção plástica, que precisam produzir uma gama de mais de 200 tarefas diferentes. Cada uma das máquinas pode processar um conjunto de tarefas, e cada item possui um conjunto de máquinas que podem processá-los. Cada tarefa corresponde a um conjunto dos mesmos itens produzidos como um.

Ressalta-se que alguns itens não podem ser processados por algumas máquinas e o tempo de processamento de um mesmo item pode variar em diferentes máquinas.

Os itens produzidos possuem demanda definida e são classificados pelas categorias A, B, C, D e E, de acordo com essa demanda.

Os itens classificados como classe A são os itens mais vendidos, e que, portanto, não podem faltar no estoque. Essa classe de itens deve ser priorizada, seguidas pelas classes B e C, que são os itens *make-to-stock* (MTS), ou seja, itens que devem ser produzidos em quantidade suficiente para serem estocados, pois possuem uma política de estoque considerando estoque de segurança e ideal.

Os itens da classe D são os classificados como *make-to-order* (MTO), e devem ser produzidos apenas após o pedido do cliente ser faturado, e, portanto, não há estoque desses itens e eles não serão considerados neste trabalho. Produtos classificados como E são os itens descontinuados, por não terem mais vendas consideráveis ou por estratégia da empresa.

3.3 PREMISSAS PARA O PROBLEMA

Como mencionado anteriormente, pretende-se neste trabalho apresentar um modelo para determinar o sequenciamento de tarefas em 24 máquinas paralelas que executam uma gama de tarefas com tempos de processamento diferentes.

O modelo a ser desenvolvido deve prezar pelo cumprimento de um objetivo básico para a empresa, ou seja, a redução do atraso de produção, procurando assim melhorar o nível de serviço. Com isso, pretende-se executar todas as tarefas ou ordens de produção dentro dos prazos determinados ou, que o atraso seja minimizado.

Para garantir o cumprimento deste objetivo será aplicada uma multa por atraso, e o objetivo do modelo será o de reduzir essa multa. O peso da multa será dado conforme a necessidade de produção, quanto maior a necessidade de produção, maior será a multa.

Assim, a partir do modelo (2.16) - (2.27), deve-se, considerando as premissas necessárias, construir um modelo que determine a sequência ótima de execução das tarefas nas máquinas que resulte na menor multa possível.

O objetivo a ser determinado para o sequenciamento da produção é a grandeza de tempo, tomando como referência a data de início da programação, ou seja, quando será feita a programação e para quando será programado o sequenciamento.

Para a obtenção dessas definições, uma série de características devem ser considerada:

- O dinamismo do setor comercial, que diariamente faz o *input* das ordens de venda durante a jornada de trabalho, que somente serão consideradas, em um relatório dos valores compilados no dia seguinte;
- A regra de prazo de saída da fábrica ou do CD para entrega de até três dias úteis a partir da data de entrada dos pedidos para os itens produzidos na empresa;

- O mercado para o qual a empresa é fornecedora exige rapidez e pontualidade na entrega dos pedidos;
- Cada tarefa considerada no sequenciamento corresponde a um conjunto de um mesmo item a ser produzido.

Os parâmetros adotados pela produção devem garantir que:

- O peso da multa deverá levar em consideração a necessidade de priorização da produção;
- Fatores que geram necessidade de priorização são: o produto estar com pedidos atrasados, ou mesmo estar próximo do prazo de entrega sem estoque suficiente para atendê-lo, seguido de itens de classe A abaixo do estoque de segurança. Outro fator a ser considerado é a necessidade apontada pelo *forecast*, que é uma previsão da demanda que o item terá para o mês subsequente;
- O sequenciamento deve considerar a menor data de entrega;
- A programação da produção será realizada semanalmente, com revisões diárias. Assim nas revisões será possível considerar os pedidos realizados no dia anterior;
- A programação deve considerar também a política de estoque que possui a quantidade de segurança e a quantidade ideal (máxima);
- Em nível de produção, os clientes da empresa têm igual importância, portanto não existe prioridade entre os clientes;
- Não há máquina para produção exclusiva de um produto, as máquinas devem ser compartilhadas entre os itens;
- As tarefas não devem permanecer por mais de um tempo determinado em uma mesma máquina que é de quinze dias (que será tratado nos dados do problema);
- Cada tarefa deve permanecer no mínimo um dia (24 horas) em máquina (que será também considerado na entrada dos dados);
- Todas as máquinas devem ser utilizadas. Não pode haver ociosidade dos recursos.

3.4 MODELAGEM PROPOSTA

Após a implementação em AMPL do exemplo apresentado na seção 2.6, procurou-se generalizar o modelo considerando-se $m = 24$ máquinas e n tarefas (conjunto de produtos) a serem produzidas, o que corresponde a situação atual da fábrica considerada no estudo de caso.

Com o intuito de apresentar um novo modelo, inicialmente procurou-se expandir o modelo (2.16)-(2.27) para o caso em que se consideram sete tarefas diferentes e duas máquinas paralelas, com tempos de processamento e prazos de entrega distintos, considerando-se as restrições já impostas naquele modelo. A escolha de uma instância pequena se deu para facilitar a etapa de validação do modelo.

Os seguintes parâmetros foram considerados:

- n : *tarefas* $\{i = 1..n\}$ - O número de tarefas a serem processadas;
- m : *máquinas* $\{k = 1..m\}$ - O número de máquinas a serem consideradas;
- p : $\{i = 1..n; k = 1..m\}$ - O tempo de processamento da tarefa i na máquina k ;
- d : $\{i = 1..n\}$ - Prazo de execução das n tarefas. Ressalta-se que o prazo de execução é independente da máquina;
- $Rate$: $\{i = 1..n\}$ - Multa por atraso na execução da produção dos n produtos. A multa também é independente da máquina.

3.4.1 Definição das variáveis

Definiu-se as seguintes variáveis:

- x_{ik} = Tempo de início, em horas, da tarefa i na máquina k a partir da hora zero, $\forall i = 1, \dots, n$ e $\forall k = 1, \dots, m$.

- $y_{ijk} = \begin{cases} 1, & \text{se a tarefa } i \text{ preceder a tarefa } j \text{ na máquina } k \\ 0, & \text{caso contrário} \end{cases}, \forall i, j = 1, \dots, n, \quad e$
 $\forall k = 1, \dots, m.$
- $z_{ik} = \begin{cases} 1, & \text{se a tarefa } i \text{ é processada na máquina } k \\ 0, & \text{caso contrário} \end{cases}; \forall i = 1, \dots, n, \forall k = 1, \dots, m.$

3.4.2 Restrições de não interferência entre as tarefas

As restrições de não interferência garantem que duas tarefas distintas i e j , com tempos de processamento na máquina k dados por p_{ik} e p_{jk} , respectivamente, não serão processadas simultaneamente na máquina k , se o tempo de início de processamento da tarefa i na máquina k é maior ou igual ao tempo de início de processamento da tarefa j acrescido do tempo de processamento da tarefa j na máquina k , que matematicamente pode ser expresso por:

$$x_{ik} \geq x_{jk} + p_{jk}, \text{ se a tarefa } i \text{ preceder } j \text{ na máquina } k, \quad (3.1)$$

ou, se o tempo de início de processamento da tarefa j na máquina k é maior ou igual ao tempo de início de processamento da tarefa i na máquina k , acrescido do tempo de processamento da tarefa i na máquina k , ou seja, se:

$$x_{jk} \geq x_{ik} + p_{ik}. \quad (3.2)$$

E para transformar essas restrições (ou-ou) em restrições simultâneas, definiu-se a variável binária auxiliar:

$$y_{ijk} = \begin{cases} 1, & \text{se } i \text{ preceder } j \\ 0, & \text{se } j \text{ preceder } i \end{cases} \text{ na máquina } k, \quad (3.3)$$

e considerou-se uma constante MM positiva grande, fazendo-se:

$$MM y_{ijk} + (x_{ik} - x_{jk}) \geq p_{jk} \quad (3.4)$$

e

$$MM(1 - y_{ijk}) + (x_{jk} - x_{ik}) \geq p_{ik}. \quad (3.5)$$

Assim, essa conversão garante que apenas uma das duas restrições (3.4) e (3.5) pode ser ativada.

3.4.3 Restrições de cobertura

Para garantir que todas as tarefas sejam processadas (cobertas), considere-se:

$$\sum_{i=1}^n \sum_{k=1}^m z_{ik} = n \quad (3.6)$$

Para garantir que todas as máquinas sejam utilizadas, considere-se:

$$\sum_{k=1}^m z_{ik} \geq 1, \quad \forall i = 1, \dots, n. \quad (3.7)$$

3.4.4 Restrições de empacotamento

Para garantir que cada tarefa seja processada apenas uma vez, considere-se:

$$\sum_{k=1}^m z_{ik} = 1, \quad \forall i = 1, \dots, n. \quad (3.8)$$

3.4.5 Restrições de cumprimento do prazo de execução

Para garantir a restrição de cumprimento de prazo de execução, considerando-se s_{jk} uma variável irrestrita, o prazo de execução da tarefa j pode ser dado por:

$$d_j = x_{jk} + p_{jk} + s_{jk}. \quad (3.9)$$

Quando $s_{jk} \geq 0$, o prazo é cumprido, e se $s_{jk} < 0$, a multa é aplicada pelo atraso.

Da mesma forma como feito na equação (2.14), considerando-se a substituição:

$$s_{jk} = s_{jk}^- - s_{jk}^+, \quad (3.10)$$

com $s_{jk}^-, s_{jk}^+ \geq 0$, então a restrição (3.4) se torna:

$$x_{jk} + s_{jk}^- - s_{jk}^+ \leq d_j - p_{kj}. \quad (3.11)$$

No entanto, a restrição (3.10) só deverá valer na máquina k quando a tarefa j for executada nesta máquina. Matematicamente, essa condição é equivalente a dizer que se,

$$z_{ik} > 0, \quad (3.12)$$

então,

$$x_{jk} + s_{jk}^- - s_{jk}^+ \leq d_j - p_{kj}. \quad (3.13)$$

As restrições (se-então) devem ser transformadas em restrições simultâneas. Para isso, considerando-se a constante $MM > 0$, e definindo-se a variável binária auxiliar:

$$yy_{ik} = \begin{cases} 1, & z_{ik} > 0 \\ 0, & \text{caso contrário} \end{cases}, \forall i = 1, \dots, n \text{ e } \forall k = 1, \dots, m, \quad (3.14)$$

pode-se escrever as seguintes restrições simultâneas:

$$x_{jk} + s_{jk}^- - s_{jk}^+ \leq d_j - p_{kj} + Myy_{ik}, \quad (3.15)$$

$$z_{ik} \leq M * (1 - yy_{ik}). \quad (3.16)$$

As restrições (3.14) e (3.15) garantem que se a tarefa j for processada na máquina k , então o tempo de processamento nesta máquina deverá ser satisfeito.

3.4.6 Multa por atraso

O custo da multa por atraso pode ser dado por:

$$\sum_{i=1}^n \sum_{k=1}^m R_i * s_{ik}^+ \quad (3.17)$$

3.4.7 Modelo algébrico considerando n tarefas e m máquinas

O modelo matemático para determinar o sequenciamento de tarefas que minimiza a multa por atraso no prazo de entrega das tarefas é então dado pelas equações (3.18-3.27).

$$\text{Minimizar } Z = \sum_{i=1}^n \sum_{k=1}^m R_i * s_{ik}^+ \quad (3.18)$$

S. a:

$$MM * y_{ijk} + (x_{ik} - x_{jk}) \geq p_{jk} \quad \forall i, j, k, i \neq j \quad (3.19)$$

$$MM * (1 - y_{ijk}) + (x_{jk} - x_{ik}) \geq p_{ik} \quad \forall i, j, k, i \neq j \quad (3.20)$$

$$x_{ik} + s_{ik}^- - s_{ik}^+ \leq d_i - p_{ik} + MM * yy_{ik} \quad \forall i, k \quad (3.21)$$

$$z_{ik} \leq MM * (1 - yy_{ik}) \quad \forall i, k \quad (3.22)$$

$$\sum_{i=1}^n z_{i,k} \geq 1 \quad \forall k \quad (3.23)$$

$$\sum_{i=1}^n \sum_{k=1}^m z_{i,k} = n \quad (3.24)$$

$$\sum_{k=1}^m z_{i,k} = 1 \quad \forall i \quad (3.25)$$

$$x_{i,k} \geq 0 \quad \forall i, k \quad (3.26)$$

$$y_{i,j,k}, yy_{ik}, z_{i,k} \in \{0,1\} \quad \forall i, j, k \quad (3.27)$$

Os dados utilizados para validação do modelo podem ser observados na Tabela 2.

Tabela 2 - Tempo de processamento para duas máquinas

Tarefas	Prazo de execução (hora)	Tempo de processamento (horas)		Multa (hora)
		Máquina 1	Máquina 2	
1	72	46.74	-	0.29
2	400	317.58	-	0.38
3	200	119.64	100.5	0.21
4	450	286.89	-	0.33
5	100	58.58	82.44	0.33
6	250	139.08	139.08	0.38
7	48	24.00	24.00	0.25

Fonte: Autoria própria

Considerou-se que no tempo de processamento utilizado está sendo considerado o tempo de preparação da máquina para realizar a tarefa a qual foi destinada.

O modelo em AMPL a ser resolvido utilizando o Solver GUROBI, pode ser observado na Figura 11.

O arquivo para a entrada de dados está representado na Figura 12. Utiliza-se um valor grande (999), muito maior que todos os outros valores utilizados como tempo de processamento, para os itens que não podem ser processados na máquina j , a fim de “proibir” seu sequenciamento em uma máquina que não pode produzi-lo.

Neste exemplo, as tarefas 1, 2 e 4 não podem ser processadas na máquina 2.

Figura 11 – Modelo generalizado em AMPL

```

-----modelo para m máquinas-----
param n;
param m;
param MM;
param p {i in 1..n, k in 1..m};
param d {i in 1..n};
param R {i in 1..n};
var x{i in 1..n, k in 1..m} ≥ 0;
var y{i in 1..n, j in 1..n, k in 1..m} binary;
var yy{i in 1..n, j in 1..n} binary;
var z {i in 1..n, k in 1..m} binary
var sM{i in 1..n} ≥ 0;
var sP{i in 1..n} ≥ 0;
Minimize PerDayPenalty: sum{i in 1..n} Rate[i] * sP[i];
subject to C1 {i in 1..n, j in 1..n, k in 1..m : i! = j}: MM * y[i, j, k] + (x[i, k] - x[j, k]) ≥ p [j, k];
subject to C2 {i in 1..n, j in 1..n, k in 1..m : i! = j} : MM * (1 - y[i, j, k]) + (x[j, k] - x[i, k]) ≥ p [i, k]
subject to C3 {i in 1..n, k in 1..m} : x[i, k] + sM [i, k] - sP[i, k] ≤ d[i] - p[i, k] + MM * yy[i, k];
subject to C4 {i in 1..n, k in 1..m} : z [i, k] ≤ M * (1 - yy[i, k]);
subject to C5 {i in 1..n}: sum {k in 1..m} z [i, k] = 1;
subject to C6 : sum{i in 1..n, k in 1..m} z [i, k] = n;
subject to C7 {k in 1..m} : sum {i in 1..n} z [i, k] ≥ 1;

```

Fonte: Autoria própria

Figura 12 – Modelo generalizado em AMPL arquivo de dados

```

----- arquivo de dados modelo para m máquinas -----
param n = 7;
param m = 2;
param MM = 100000;
param p: 1 2 :=;
1  46.74  999
2  317.58 999
3  119.64 100.5
4  286.89 999
5   58.58 82.44
6  139.08 139.08
7   24.00 24.00;
param d: 1 2 3 4 5 6 7 8 := 1 72 400 200 450 100 250 48;
param Rate : 1 2 3 4 5 6 7 8 := 1 0.29 0.38 0.21 0.33 0.33 0.38 0.25;

```

Fonte: Autoria própria

3.5 ESTRATÉGIA DE SOLUÇÃO

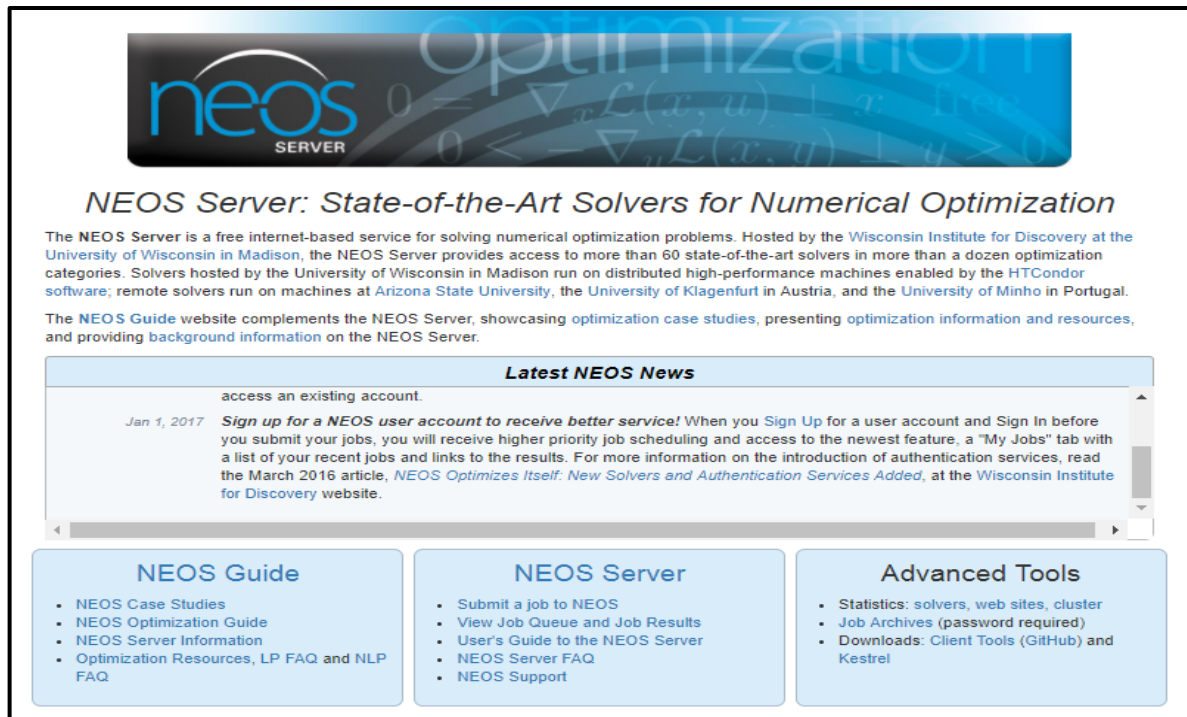
Nesta etapa, como suporte para a resolução do problema, utilizou-se o site *NEOS Server: State-of-Art Solver for Numerical Optimization* (www.neos-server.org). Trata-se de um *site* gratuito para solução de problemas de otimização numérica sem limite de variáveis. Com suporte da *University of Wisconsin*, em Madison, os modelos matemáticos são executados remotamente em máquinas de alto desempenho distribuídos em várias universidades pelo mundo como por exemplo na *Arizona State University*, *University of Klagenfurt* na *Austria*, e *University of Minho* em Portugal (NEOS SERVER GUIDE, 2014).

A seguir são apresentadas as etapas, explicadas passo a passo, de como executar um modelo matemático remotamente no NEOS Server.

Primeiramente, basta acessar o endereço www.neos-server.org/neos/ e clicar no link 'submit a Job to NEOS', como mostrado na Figura 13.

A etapa seguinte compreende a definição do tipo de problema representado pelo modelo matemático, para determinar a escolha do *solver* a ser utilizado na resolução do problema bem como a linguagem em que o modelo será escrito.

Figura 13 – Página inicial do site



neos
SERVER

Optimization

NEOS Server: State-of-the-Art Solvers for Numerical Optimization

The NEOS Server is a free internet-based service for solving numerical optimization problems. Hosted by the Wisconsin Institute for Discovery at the University of Wisconsin in Madison, the NEOS Server provides access to more than 60 state-of-the-art solvers in more than a dozen optimization categories. Solvers hosted by the University of Wisconsin in Madison run on distributed high-performance machines enabled by the HTCondor software; remote solvers run on machines at Arizona State University, the University of Klagenfurt in Austria, and the University of Minho in Portugal.

The NEOS Guide website complements the NEOS Server, showcasing optimization case studies, presenting optimization information and resources, and providing background information on the NEOS Server.

Latest NEOS News

access an existing account.

Jan 1, 2017 **Sign up for a NEOS user account to receive better service!** When you Sign Up for a user account and Sign In before you submit your jobs, you will receive higher priority job scheduling and access to the newest feature, a "My Jobs" tab with a list of your recent jobs and links to the results. For more information on the introduction of authentication services, read the March 2016 article, *NEOS Optimizes Itself: New Solvers and Authentication Services Added*, at the Wisconsin Institute for Discovery website.

NEOS Guide

- NEOS Case Studies
- NEOS Optimization Guide
- NEOS Server Information
- Optimization Resources, LP FAQ and NLP FAQ

NEOS Server

- Submit a job to NEOS
- View Job Queue and Job Results
- User's Guide to the NEOS Server
- NEOS Server FAQ
- NEOS Support

Advanced Tools

- Statistics: solvers, web sites, cluster
- Job Archives (password required)
- Downloads: Client Tools (GitHub) and Kestrel

Fonte: www.neos-server.org (Acesso em: 22/10/2017)

Na Figura 14 podem ser observados alguns tipos de formas de modelagem no NEOS Server, como por exemplo, PL, PNL, PLI e PLIM, entre outras.

Figura 14 – Formas de modelagem disponíveis no NEOS Server



Job Queue Tools	+
Application	+
Bound Constrained Optimization	+
Combinatorial Optimization and Integer Programming	+
Complementarity Problems	+
Extended Mathematical Programming	+
Global Optimization	+
Linear Network Programming	+
Linear Programming	+
Mathematical Programs with Equilibrium Constraints	+
Mixed Integer Linear Programming	+
Mixed Integer Nonlinearly Constrained Optimization	+
Mixed-Integer Optimal Control Problems	+
Nondifferentiable Optimization	+
Nonlinearly Constrained Optimization	+
Second Order Conic Programming	+
Semi-infinite Optimization	+
Semidefinite Programming	+
Stochastic Linear Programming	+

Fonte: www.neos-server.org (acesso em: 22/10/2017)

Na Figura 15 podem ser observados os *solvers* disponíveis no NEOS Server.

Após a classificação do tipo de problema e a escolha do *Solver a ser considerado*, é necessário inserir os arquivos para resolução do problema, como é mostrado na Figura 16.

Figura 15 – Alguns solvers disponíveis NEOS Server

<p>Linear Programming</p> <ul style="list-style-type: none"> • BDMLP [GAMS Input] • bpmpl [AMPL Input][LP Input][MPS Input][QPS Input] • Clp [MPS Input] • CPLEX [AMPL Input][GAMS Input][LP Input][MPS Input] • FICO-Xpress [AMPL Input][GAMS Input][MOSEL Input][MPS Input] • Gurobi [AMPL Input][GAMS Input][LP Input][MPS Input] • MOSEK [AMPL Input][GAMS Input][LP Input][MPS Input] • OQOP [AMPL Input] • SoPlex80bit [LP Input][MPS Input]
<p>Mathematical Programs with Equilibrium Constraints</p> <ul style="list-style-type: none"> • filterMPEC [AMPL Input] • Knitro [GAMS Input] • NLPEC [GAMS Input]
<p>Mixed Integer Linear Programming</p> <ul style="list-style-type: none"> • Cbc [AMPL Input][GAMS Input][MPS Input] • CPLEX [AMPL Input][GAMS Input][LP Input][MPS Input] • feaspump [AMPL Input][CPLEX Input][MPS Input] • FICO-Xpress [AMPL Input][GAMS Input][MOSEL Input][MPS Input] • Gurobi [AMPL Input][GAMS Input][LP Input][MPS Input] • MINTO [AMPL Input] • MOSEK [AMPL Input][GAMS Input][LP Input][MPS Input] • proxy [CPLEX Input][MPS Input] • qsopl_ex [AMPL Input][LP Input][MPS Input] • scip [AMPL Input][CPLEX Input][GAMS Input][MPS Input][OSIL Input][ZIMPL Input] • SYMPHONY [MPS Input]
<p>Mixed Integer Nonlinearly Constrained Optimization</p> <ul style="list-style-type: none"> • AlphaECP [GAMS Input] • BARON [AMPL Input][GAMS Input] • Bonmin [AMPL Input][GAMS Input] • Couenne [AMPL Input][GAMS Input] • DICOPT [GAMS Input] • FilMINT [AMPL Input] • Knitro [AMPL Input][GAMS Input] • LINDOGlobal [GAMS Input] • MINLP [AMPL Input] • SBB [GAMS Input] • scip [AMPL Input][CPLEX Input][GAMS Input][MPS Input][OSIL Input][ZIMPL Input]

Fonte: www.neos-server.org (acesso em: 22/10/2017)

Para a inserção dos dados são necessários três arquivos, a saber:

- O primeiro, *Model File* (um arquivo de extensão *.mod), com o modelo matemático na linguagem escolhida;
- O segundo, *Data (*.dat)*, com a entrada dos parâmetros (Figura 15);
- E o último, *Commands File (*.dat)*, é o conjunto de comandos que especifica o que o usuário deseja visualizar.

Figura 16 – *Input* dos dados

Web Submission Form

Model File
Enter the location of the AMPL model (local file)
 Nenhum arquivo selecionado

Data File
Enter the location of the AMPL data file (local file)
 Nenhum arquivo selecionado

Commands File
Enter the location of the AMPL commands file (local file)
 Nenhum arquivo selecionado

Comments

Additional Settings
 Dry run: generate job XML instead of submitting it to NEOS
 Short Priority: submit to higher priority queue with maximum CPU time of 5 minutes
E-Mail address:

Please do not click the 'Submit to NEOS' button more than once.

By submitting a job, you have accepted the [Terms of Use](#)

Fonte: www.neos-server.org (acesso em: 22/10/2017)

Após o *upload* dos três arquivos e a introdução de um endereço de *e-mail* válido, basta esperar o servidor devolver os dados solicitados, ou enviar alguma mensagem de erro, quando for o caso.

O servidor devolve como resposta dados como o número de iterações, número de variáveis e restrições além das respostas esperadas.

Utilizou-se o *solver* GUROBI entre os disponíveis para problemas de programação linear inteira e mista, e o modelo matemático (Figura11) foi escrito em AMPL (*A Mathematical Programming Language*).

3.6 VALIDAÇÃO DO MODELO

Primeiramente, com o intuito de validar o modelo, executou-se o modelo apresentado na Figura 11, com os dados da Tabela 2 da seção 3.3.7 deste trabalho. Depois passou-se a aplicar o modelo em alguns cenários reais de tamanho reduzido.

Após processar o modelo da Figura 11, com os dados indicados na Figura 12, o servidor devolve as seguintes informações de execução:

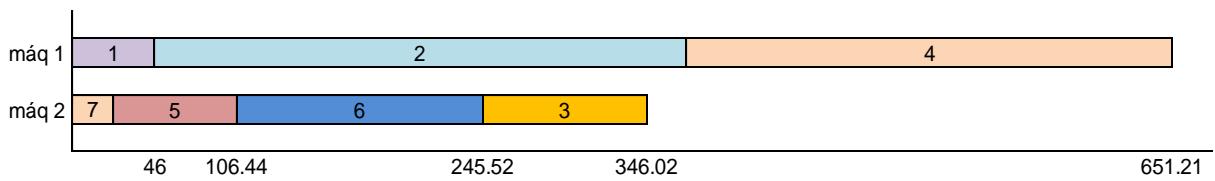
- Foram necessárias 664 iterações simplex;
- O uso de 1 nó *Branch-and-bound*;
- O modelo considerou 154 variáveis ao todo das quais 112 são variáveis binárias e 42 são lineares;
- 206 restrições lineares das quais 22 são de igualdade e 184 de desigualdade.

A solução apresentou a sequência 1 → 2 → 4 de tarefas a serem processadas na máquina 1 e a sequência de tarefas 7 → 5 → 6 → 3 a serem processadas na máquina 2, com uma multa total de 99,19. Observa-se um atraso no processamento da tarefa 4 na máquina 1, e no processamento das tarefas 3 e 5 na máquina 2.

Como pode-se observar na Figura 17, que apresenta o sequenciamento das tarefas, a tarefa 4 terminou em 651,21 horas, quando o prazo da mesma era de 450 horas, portanto o atraso foi de 201,21 horas e a multa resultante deste atraso é de $201,21 * 0,33 = 66,40$.

A tarefa 5 tem prazo entrega de 100 horas e foi finalizada em 106,44, apresentando assim um atraso de 6,44 horas, e resultando, portanto, uma multa de $6,44 * 0,33 = 2,13$. Já a tarefa 3 tem prazo de 200 horas e foi finalizada em 346,02, ocasionando portanto um atraso de 146,02 horas e gerando uma multa de $146,02 * 0,21 = 30,66$.

Figura 17 – Sequenciamento máquinas



Fonte: Autoria própria

Somando as multas geradas por tarefa nas máquinas 1 e 2 temos $66,40 + 2,13 + 30,66 = 99,19$. Para continuar com a validação do modelo foram propostos cenários, a fim de se observar se os resultados obtidos eram os esperados.

3.6.1 Cenário 1

Considera-se nesse cenário, que apenas uma tarefa possa ser processada na máquina 2, e as demais, somente na máquina 1.

Tabela 3 - Tempo de processamento para duas máquinas cenário 1

Tarefas	Prazo de execução (hora)	Tempo de processamento (horas)		Multa (hora)
		Máquina 1	Máquina 2	
1	72	46.74	-	0.29
2	400	317.58	-	0.38
3	200	119.64	-	0.21
4	450	286.89	-	0.33
5	100	58.58	-	0.33
6	250	139.08	-	0.38
7	48	24.00	24.00	0.25

Fonte: Autoria própria

Espera-se, portanto, que o modelo retorne uma solução que indique um sequenciamento em que a máquina 2 processe somente esta tarefa, e as demais sejam sequenciadas para processamento na máquina 1. Assim, para os dados da Tabela 3, espera-se que somente a tarefa 7 seja executada na máquina 2, e todas as

outras sejam processadas na máquina 1. O arquivo de dados para esse cenário pode ser observado a seguir na Figura 18.

Figura 18 – Modelo generalizado em AMPL cenário 1

```

----- arquivo de dados modelo para m máquinas -----
param n = 7;
param m = 2;
param MM = 100000;
param p : 1 2 :=;
1  46.74  999
2  317.58  999
3  119.64  999
4  286.89  999
5   58.58  999
6  130.08  999
7   24.00  24.00;
param d : 1 2 3 4 5 6 7 8 := 1 72 400 200 450 100 250 48;
param Rate : 1 2 3 4 5 6 7 8 := 1 0.29 0.38 0.21 0.33 0.33 0.38 0.25;

```

Fonte: Autoria própria

Adota-se um número grande como tempo de processamento para as tarefas que não podem ser produzidas por uma máquina como forma de proibir que elas sejam selecionadas (aumentará muito o custo da multa). Após executar o modelo com os dados da Figura 18, obtém-se uma solução com multa resultante de 314,33, em que somente a tarefa 7 é realizada na máquina 2, como era esperado.

As demais tarefas são realizadas na máquina 1, seguindo a sequência 1 → 5 → 6 → 3 → 2 → 4, como indicado na Figura 19. Observa-se que as tarefas 2, 3, 4 e 5 na máquina 1 tiveram atraso, que resultaram na multa apresentada.

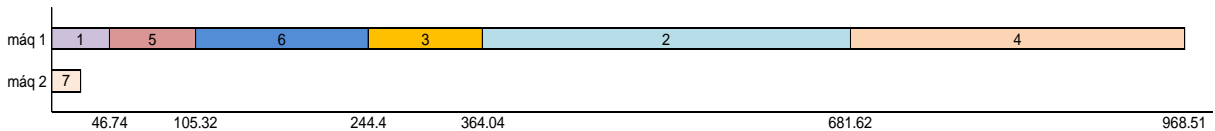
A sequência obtida na máquina 1 e a tarefa destinada para a máquina 2 e o tempo de início e fim das tarefas podem ser observadas na Figura 19.

Para executar o modelo com os dados deste cenário foram necessárias:

- 154 variáveis, das quais 112 são binárias e 42 lineares;

- 206 restrições ao todo em que 22 são de igualdade e 184 de desigualdade;
- 1.459 iterações simplex;
- 120 nós *branch-and-bound*.

Figura 19 – Sequenciamento máquinas cenário 1



Fonte: Autoria própria

3.6.2 Cenário 2

Para esse cenário considera-se que todas as tarefas a serem processadas na máquina 2, tem um tempo de processamento muito maior do que na máquina 1. Os tempos de processamento de todas as tarefas na máquina 2 são muito maiores do que na máquina 1.

Espera-se com este cenário verificar se a restrição de exigência de utilização de todas está sendo cumprida. Os dados considerados para este cenário podem ser observados na Tabela 4. Assim, adota-se o tempo de processamento de 600 horas, que é maior do que o tempo máximo de processamento por tarefa permitido, ou seja, de 15 dias (360 horas).

Tabela 4 - Tempo de processamento para duas máquinas cenário 2

Tarefas	Prazo de execução (hora)	Tempo de processamento (horas)		Multa (hora)
		Máquina 1	Máquina 2	
1	72	46.74	600	0.29
2	400	317.58	600	0.38
3	200	119.64	600	0.21
4	450	286.89	600	0.33
5	100	58.58	600	0.33
6	250	139.08	600	0.38
7	48	24.00	600	0.25

Fonte: Autoria própria

Espera-se que, se a regra de que todas as máquinas devem ser utilizadas for cumprida, que pelo menos uma das tarefas deve ser executada na máquina 2, mesmo que os tempos de processamento sejam maiores. O arquivo de dados para esse cenário pode ser observado na Figura 20.

Figura 20 – Modelo generalizado em AMPL cenário 2

```

----- arquivo de dados modelo para m máquinas -----
param n = 7;
param m = 2;
param MM = 100000;
param p : 1 2 :=;
1  46.74  600
2  317.58  600
3  119.64  600
4  286.89  600
5   58.58  600
6  130.08  600
7   24.00  600;
param d : 1 2 3 4 5 6 7 8 := 1 72 400 200 450 100 250 48;
param Rate : 1 2 3 4 5 6 7 8 := 1 0.29 0.38 0.21 0.33 0.33 0.38 0.25;

```

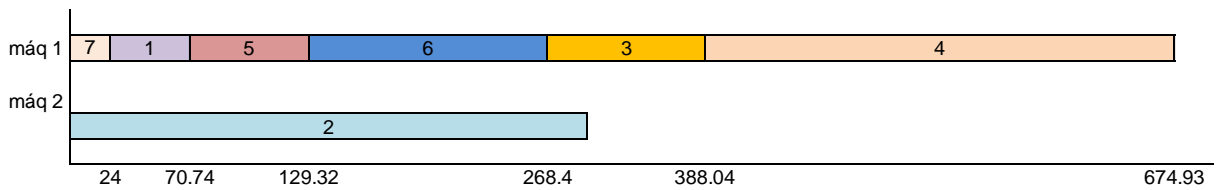
Fonte: Autoria própria

A solução determinou que apenas a tarefa 2 será processada na máquina 2, gerando uma multa de 206,38.

Apesar da multa aplicada por atraso na tarefa 3 ser menor do que a multa aplicada por atraso para a tarefa 2, o prazo de execução da tarefa 2 é maior e, portanto, o atraso será menor que se fosse feita a tarefa 3, conseqüentemente a multa será menor. As demais tarefas foram atribuídas à máquina 1, obedecendo a seqüência 7 → 1 → 5 → 6 → 3 → 4. Sofrem atraso as tarefas 3, 4, 5 e 6 na máquina 1 e a tarefa 2 na máquina 2.

Na Figura 21 é apresentada a seqüência apontada para a máquina 1 e a tarefa alocada para a máquina 2, além dos tempos de início e fim das tarefas.

Figura 21 – Sequenciamento máquinas cenário 2



Fonte: Autoria própria

Para determinar essa solução foram utilizadas:

- 154 variáveis, das quais 112 são binárias e 42 lineares;
- 206 restrições ao todo em que 22 são de igualdade e 184 de desigualdade;
- 3.107 iterações simplex;
- 199 nós *branch-and-bound*.

3.6.3 Cenário 3

Neste cenário as tarefas 1 a 4 só podem ser processadas na máquina 1, e as demais, na máquina 2. Na Tabela 5 são apresentados os dados utilizados no cenário 3.

Tabela 5 - Tempo de processamento para duas máquinas cenário 3

Tarefas	Prazo de execução (hora)	Tempo de processamento (horas)		Multa (hora)
		Máquina 1	Máquina 2	
1	72	46.74	-	0.29
2	400	317.58	-	0.38
3	200	119.64	-	0.21
4	450	286.89	-	0.33
5	100	-	82.44	0.33
6	250	-	139.08	0.38
7	48	-	24.00	0.25

Fonte: Autoria própria

O esperado para este cenário é que a solução obtida apresente um sequenciamento que leve isso em consideração, ou seja, que as tarefas 1, 2, 3 e 4

devem ser alocadas na máquina 1, e as tarefas 5, 6 e 7 devem ser alocadas na máquina 2. O arquivo de dados é apresentado na Figura 22.

Figura 22 – Modelo generalizado em AMPL cenário 3

```

----- arquivo de dados modelo para m máquinas -----
param n = 7;
param m = 2;
param MM = 100000;
param p : 1 2 :=;
1  46.74  999
2  317.58  999
3  119.64  999
4  286.89  999
5  999     82.44
6  999     139.08
7  999     24.00;
param d: 1 2 3 4 5 6 7 8 := 1 72 400 200 450 100 250 48;
param Rate : 1 2 3 4 5 6 7 8 := 1 0.29 0.38 0.21 0.33 0.33 0.38 0.25;

```

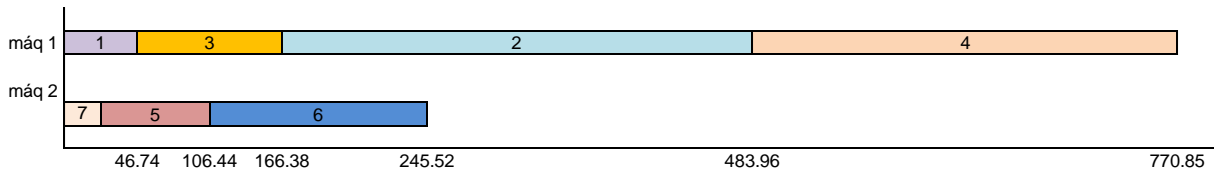
Fonte: Autoria própria

Como esperado, as tarefas 1, 2, 3 e 4 foram alocadas na máquina 1, na sequência 1 → 3 → 2 → 4. E as tarefas 5, 6 e 7 foram alocadas na máquina 2, na sequência 7 → 5 → 6. A multa obtida neste cenário foi de 139.91. A Figura 23 mostra a sequência de tarefas alocadas para as máquinas 1 e 2, bem como o tempo de início e fim de processamento das tarefas.

Para determinar essa solução foram utilizadas:

- 154 variáveis, das quais 112 são binárias e 42 lineares;
- 206 restrições ao todo em que 22 são de igualdade e 184 de desigualdade;
- 365 iterações simplex;
- 1 nó *branch-and-bound*.

Figura 23 – Sequenciamento máquinas cenário 3



Fonte: Autoria própria

Após a análise dos resultados obtidos em todos os cenários e comparando com o caso reduzido, tem-se como certo que todos os resultados esperados foram alcançados e, portanto, o modelo está validado e é então possível passar à aplicação do modelo para os dados do estudo de caso.

Na próxima seção será considerado um arquivo de dados para $m = 24$ máquinas e $n = 59$ tarefas, que é o estudo de caso deste trabalho.

3.7 APLICANDO O MODELO PARA $m = 24$ MÁQUINAS E $n = 59$ TAREFAS

O estudo de caso considerado neste trabalho possui $n = 59$ tarefas a serem processadas em $m = 24$ máquinas paralelas com diferentes tempos de processamento para a mesma tarefa.

Ressalta-se aqui, que se considerou como premissa para o modelo, que cada tarefa constitui um conjunto de itens do mesmo tipo, para que cada máquina pudesse processar um número mínimo de itens do mesmo tipo antes de ser preparada para processar outro tipo de item.

Assim, para as 59 tarefas consideradas aqui para sequenciamento nas 24 máquinas, tem-se um total de 214.266 itens.

A tarefa 1, por exemplo, produz um conjunto de 15.000 itens, a tarefa 2 produz 144 itens e assim por diante, Esse detalhamento de número de itens por tarefa é apresentado na tabela abaixo

Tabela 6 – Quantidade de itens por tarefa

tarefa	quantidade de itens	tarefa	quantidade de itens
1	15000	31	570
2	144	32	5000
3	493	33	2285
4	1500	34	950
5	1550	35	700
6	5000	36	500
7	11900	37	7000
8	4500	38	800
9	22050	39	2000
10	845	40	3500
11	1540	41	325
12	4500	42	3000
13	4550	43	950
14	1000	44	490
15	7000	45	500
16	5000	46	700
17	10000	47	125
18	475	48	7000
19	12700	49	160
20	5000	50	4500
21	9000	51	2009
22	9000	52	1100
23	3500	53	2470
24	990	54	770
25	7000	55	480
26	500	56	1500
27	8820	57	1000
28	2000	58	800
29	3000	59	4025
30	500		

Fonte: Autoria Própria

Aplicou-se o modelo para $n = 59$ tarefas por ser a quantidade de tarefas utilizada no plano de produção atual da empresa em estudo, assim pode ser feita a análise comparativa da situação atual com o resultado obtido no modelo. Essa análise é apresentada no Capítulo 4 deste trabalho.

Os dados utilizados para a implementação do modelo encontram-se na Tabela 6 e 7, no Anexo A.

Para a aplicação do modelo matemático nessa configuração foram necessárias:

- 89.208 variáveis, das quais 84.960 são binárias e 4.248 lineares;
- 167.172 restrições ao todo em que 1.476 são de igualdade e 165.696 de desigualdade;
- 1.710.841 iterações simplex;
- 13.016 nós *branch-and-bound*.

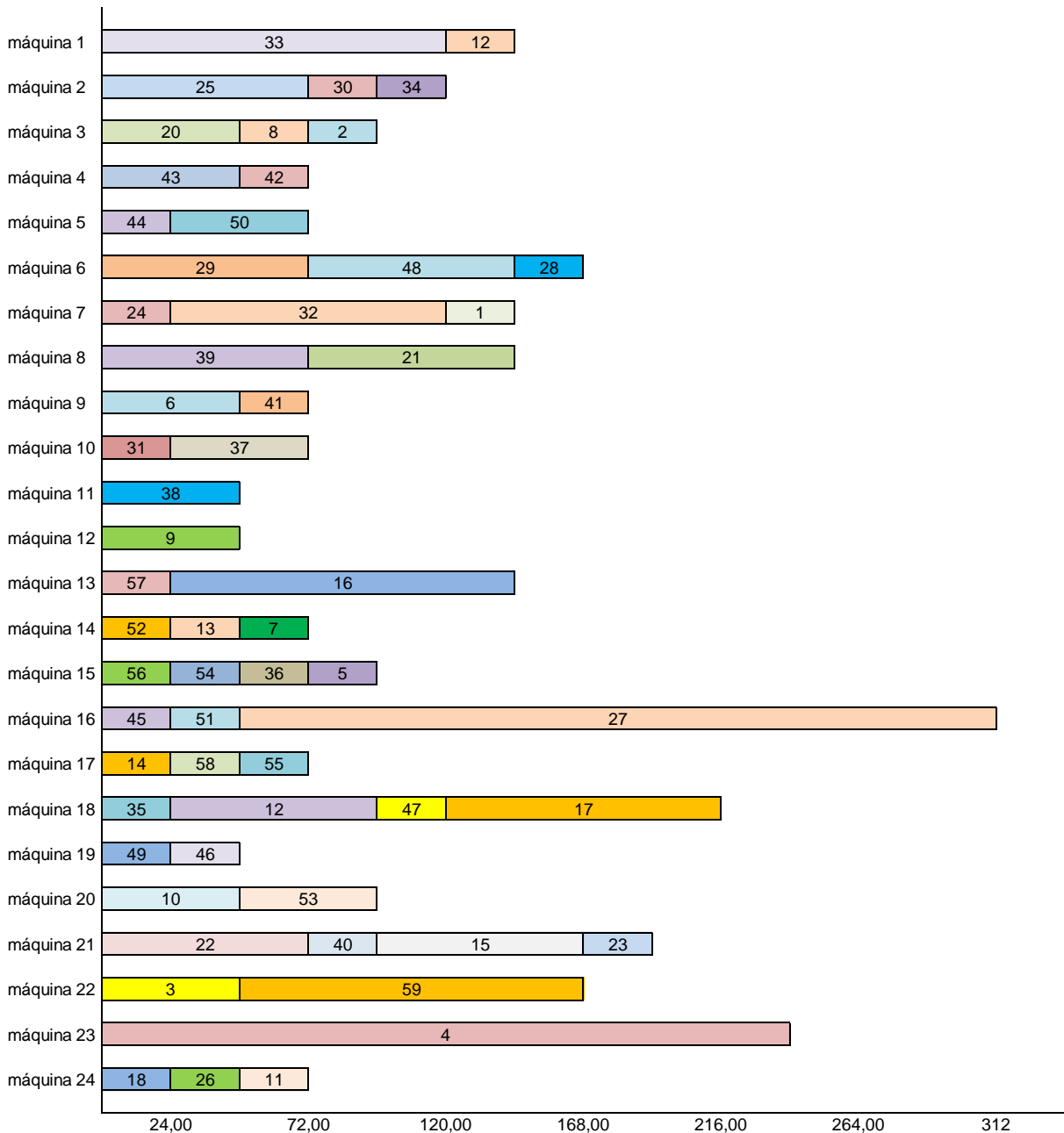
Na Figura 24 é apresentado o sequenciamento por máquina obtido para o problema de 24 máquinas e 59 tarefas, que são um conjunto de conexões, a sequência obtida é viável dentro das restrições consideradas para o problema. Na máquina 1 são processadas as tarefas 12 e 33 na sequência 33 → 12.

O processamento da primeira tarefa, a tarefa 33 inicia-se em 0 horas e é finalizado em 120 horas, quando então é iniciado o processamento da segunda tarefa, a tarefa 12, que é então finalizada em 48 horas depois, ou seja, em 168 horas. Observa-se que o prazo das duas tarefas é cumprido e, portanto, nenhuma multa é aplicada para as atividades dessa máquina.

Já na máquina 2, são processadas as tarefas 25, 30 e 34, obedecendo a sequência 25 → 30 → 34. A tarefa 25 é iniciada em 0 horas e é finalizada em 72 horas. A tarefa seguinte é iniciada em 72 e é finalizada em 96 horas, sendo, portanto, cumprindo prazo de entrega. Já a última tarefa, a 34, é iniciada em 96 horas e é finalizada em 120 horas. No entanto, como o seu prazo de entrega era de 72 horas, é aplicada a multa de 0,33 em cima do atraso de 48 horas, resultando em uma multa de $48 * 0,33 = 15,84$.

A multa gerada pela soma dos atrasos nas máquinas é de 93,60. Sofreram atraso as tarefas 34, 28, 1, 5, 47 e 17. A tarefa 28 na máquina 6, é finalizada em $x = 168$ horas. Porém seu prazo de processamento era de 96 horas e, portanto, a multa aplicada para a tarefa 28 é de $72 * 0,33 = 23,76$. A tarefa 1 na máquina 7, é finalizada em $x = 144$ horas. Como o seu prazo era 48 horas, gerando uma multa por atraso de $96 * 0,17 = 16,32$.

Figura 24 – Sequenciamento para 24 máquinas



Fonte: Autoria própria

Para a tarefa 5 na máquina 15 o prazo era de 48 horas, porém a mesma foi concluída em $x = 96$ horas, resultando em uma multa de $48 * 0,33 = 15,84$. Na máquina 18 ocorre atraso em 2 tarefas. A tarefa 47 atrasa 48 horas o que resulta em uma multa de $48 * 0,29 = 13,92$. Já a tarefa 17, atrasa 24 horas, acarretando em uma multa de $24 * 0,33 = 7,92$.

Essas são as tarefas que sofreram com atraso no processamento. A multa final se dá pela soma das multas resultantes do atraso na produção das tarefas: $15,84 + 23,76 + 16,32 + 15,84 + 13,92 + 7,92 = 93,60$.

No próximo capítulo será apresentada a análise dos resultados.

4 RESULTADOS E ANÁLISES COMPARATIVAS

Neste capítulo serão abordados os resultados obtidos com a implementação do método proposto.

Inicialmente, será detalhado o cenário atual da empresa e em seguida os resultados com o uso da otimização do sequenciamento com modelo exato. Além disso será realizada uma análise comparativa entre a sequência obtida com o uso do método e o realizado atualmente na empresa.

4.1 CENÁRIO ATUAL

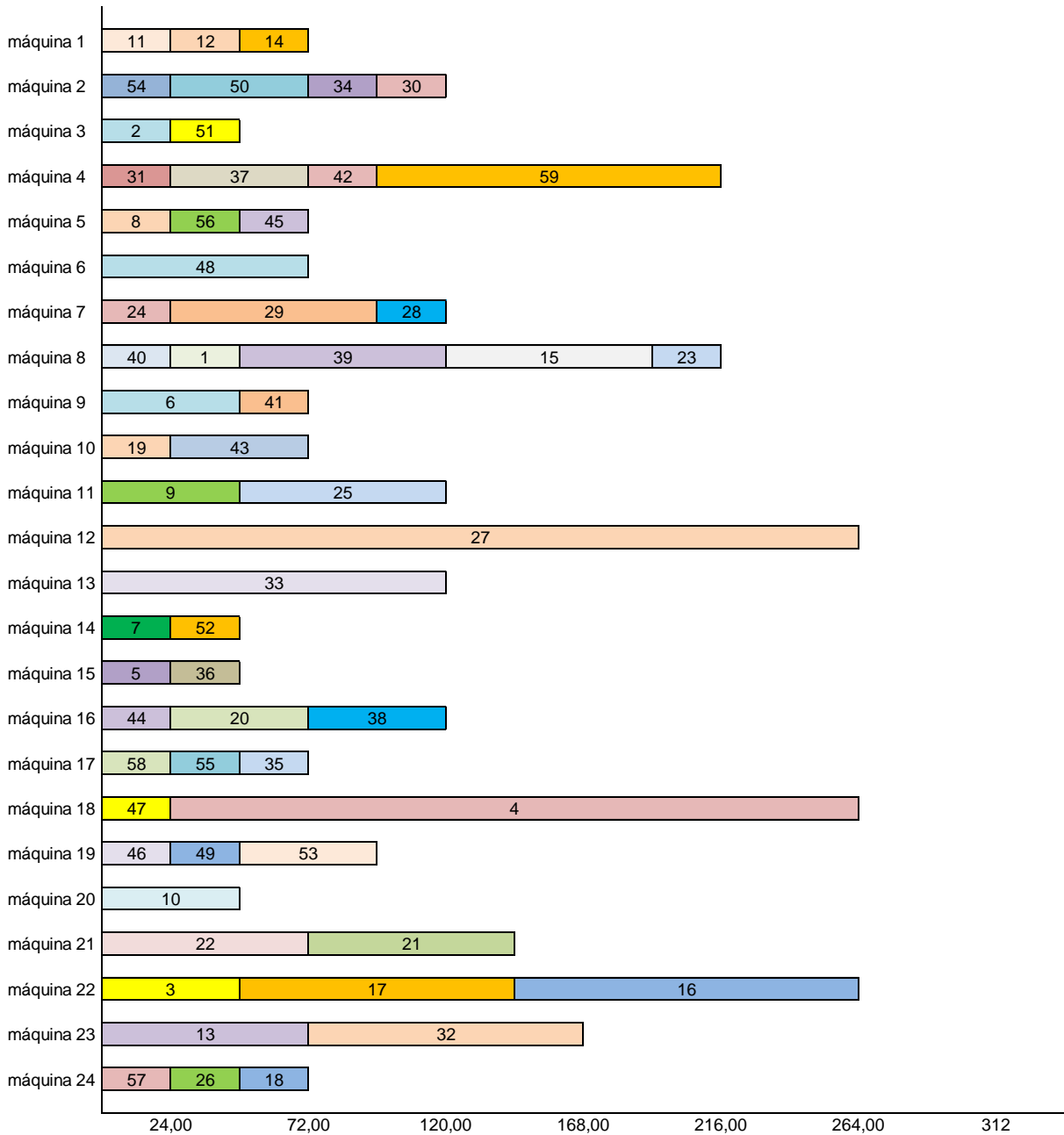
A empresa atualmente conta com um total de 24 máquinas e um portfólio de aproximadamente 300 produtos *make-to-order* (produzidos apenas após o pedido do cliente) e aproximadamente 200 produtos *make-to-stock* (produzidos para estocagem).

Este trabalho trata os itens *make-to-stock*, pois são itens que possuem política de estoque, com estoque de segurança e estoque máximo. A programação da produção é realizada com uso de planilhas de Excel, com equações simples, onde o programador trabalha incluindo tarefa a tarefa a mão, sem uso de qualquer ferramenta de otimização, sendo assim uma maneira paliativa de fazê-lo.

A programação é realizada uma vez por semana e há revisões diárias da necessidade de produção. Como a programação é feita de maneira manual o processo de programação semanal e até mesmo o processo de revisão diário demandam um tempo muito grande do programador, e não há garantia de que a sequência é a que melhor irá atender as necessidades de produção.

A sequência das tarefas obtida pela programação atual da empresa é apresentada na figura a seguir.

Figura 25 – Sequencia atual



Fonte: Autoria própria

As seqüências obtidas pela configuração atual e com o uso do modelo otimizado são totalmente diferentes. Apenas o sequenciamento da máquina 9 permaneceu igual nas duas configurações.

4.2 DESEMPENHO DO SEQUENCIAMENTO

Considerando as mesmas multas e prazos para o plano realizado atualmente pela empresa a multa resultante do atraso da produção das tarefas resulta em 104,90, ou seja, a otimização reduziu o atraso da produção em aproximadamente 11%. A seguir no Gráfico 1 são representadas as multas resultantes no cenário atual da empresa de 104,90 e a multa resultante da aplicação do modelo de 93,60.

Gráfico 1- Multa: cenário real e resultado obtido



Fonte: Autoria própria

O sequenciamento utilizado pela empresa é realizado de forma totalmente manual e a empresa não considera nenhuma ferramenta de otimização, portanto a melhora nos resultados era esperada.

4.3 ANÁLISE DA COMPLEXIDADE DO MODELO

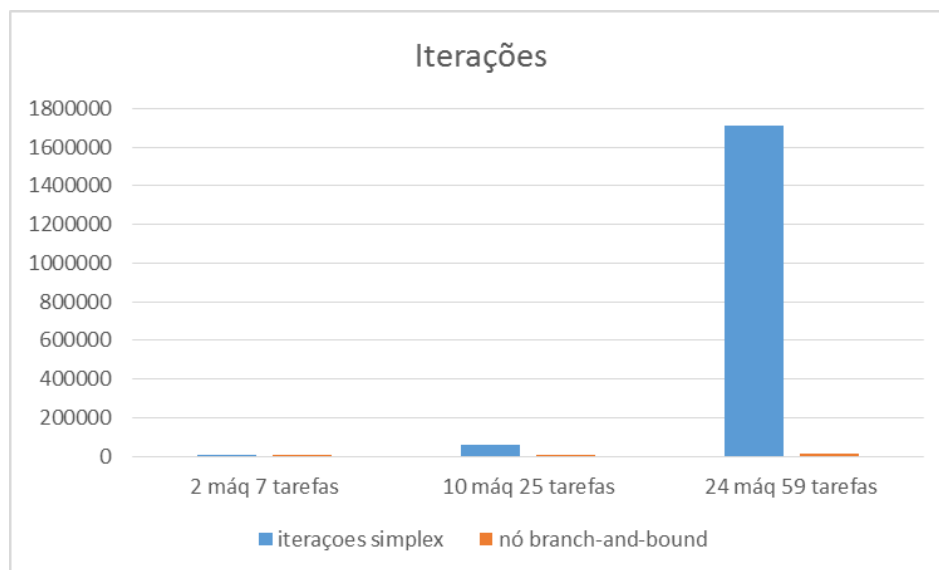
Com o intuito de analisar a complexidade do modelo à medida que o número de máquinas e tarefas cresce, buscou-se analisar as informações de números de iterações simplex, número de nós *branch-and-bound* necessários, número de variáveis e número de restrições necessários para o caso apresentado na validação

do modelo, ou seja, para o caso de 2 máquinas e 7 tarefas, para o caso do estudo de caso, de 24 máquinas e 59 tarefas e um caso intermediário de 10 máquinas e 25 tarefas dentro dos valores reais.

O caso intermediário foi aplicado com o intuito de construir a análise do aumento da complexidade do modelo. Nesse caso foram necessárias 54.545 iterações, 2.987 variáveis das quais 2.899 variáveis binárias e 88 variáveis lineares, e 3.698 restrições.

A seguir no Gráfico 2 é apresentado o comportamento do número de iterações de acordo com o cenário utilizado. Conforme o crescimento no número de máquinas e tarefas há um crescimento exponencial do número de iterações.

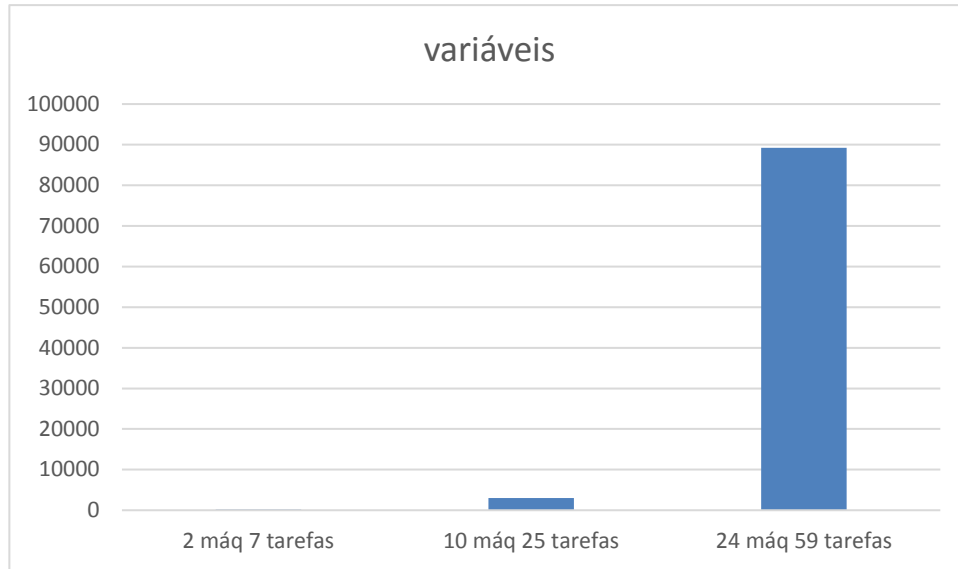
Gráfico 2- Número de iterações



Fonte: Autoria própria

No Gráfico 3 é possível ver a relação entre o número de variáveis para os diferentes cenários tratados neste trabalho. O mesmo crescimento exponencial pode ser percebido neste caso.

Gráfico 3 – Número de variáveis



Fonte: Autoria própria

No Gráfico 4 apresenta-se a relação quanto ao número de restrições necessárias para resolver cada um dos cenários. Há novamente um crescimento exponencial do número de restrições necessárias para o modelo.

Gráfico 4 - Número de restrições



Fonte: Autoria própria

Como é possível observar nos Gráficos1 a 4, a complexidade do modelo é bastante alterada conforme há um aumento no número de máquinas ou no número de tarefas.

No próximo capítulo serão apresentadas as considerações finais do trabalho e sugestões para trabalhos futuros.

5 CONSIDERAÇÕES FINAIS

Neste capítulo serão apresentadas as considerações finais e análise do trabalho, e considerações para trabalhos futuros.

Foi proposto neste trabalho um modelo matemático para resolver problemas de sequenciamento de tarefas em várias máquinas paralelas com tempos de processamento diferentes, objetivando reduzir a penalidade por atraso na produção. O modelo de programação linear inteira considerado foi implementado em linguagem AMPL utilizando o solver GUROBI processado remotamente no *Neos Server*.

A maioria dos trabalhos acadêmicos utilizam métodos heurísticos, que apesar de terem um tempo computacional menor que o método exato, e não garantem a resposta ótima, para a resolução de problemas de grande porte. Assim a contribuição de um modelo exato, que garante a certeza da otimização, é bastante relevante.

Definiu-se como medida de desempenho a minimização da penalidade gerada pelo atraso na produção e considerou-se restrições de não-interferência das tarefas, para que não seja possível a sobreposição de tarefas em uma mesma máquina, restrições de prazo de execução, restrições de cobertura, tanto de itens como das máquinas, para garantir que todos os itens sejam produzidos e que todas as máquinas sejam utilizadas.

A programação considerada pela empresa em estudo é realizada manualmente, em planilhas de *Excel*, alocando-se item a item nas máquinas possíveis e leva em consideração apenas as quantidades a serem produzidas e o conhecimento prévio do programador. Nenhum processo otimizado é utilizado para a programação.

O modelo de otimização proposto para determinar o sequenciamento das tarefas em máquinas paralelas para o caso estudado resultou em uma diminuição de 11% na multa por atraso na produção. Pensando-se economicamente e em qualidade de nível de serviço entregue ao cliente, a diminuição é bastante significativa.

5.1 DESDOBRAMENTOS E TRABALHOS FUTUROS

Para trabalhos futuros sugere-se o estudo das propriedades estruturais e a implementação de problemas de programação da produção em outros ambientes, como por exemplo, máquinas no modelo paralelas idênticas ou máquinas no modelo *job shop*, incluindo restrições diferentes das tratadas nesta pesquisa, como *setup* independente da sequência, janelas de entrega e de minimização simultânea por antecipação e atraso, com janela de tempo de produção.

É possível também considerar modelos que garantam condições de precedência, ou seja, quando for necessário que se tarefa i precede a tarefa j , então a tarefa k deve preceder a tarefa m . Ou considerar restrições que assegurem que as tarefas devem ser processadas em mais de uma máquina para que sejam concluídas.

É possível considerar outras medidas de desempenho a serem otimizadas ou até mesmo várias medidas. Sugere-se também o uso de outras ferramentas para otimizar o sequenciamento das tarefas nas máquinas.

REFERÊNCIAS BIBLIOGRÁFICAS

ARAUJO, J. S.; FUCHIGAMI, H. Y. **Mathematical modeling of the production-sequencing problem with dependent setup times and philosophy just in time.** *Exacta*, Belo Horizonte, v. 6, n. 2, p. 165-175, 2013.

ARENALES, M. et al. **Pesquisa Operacional para Cursos de Engenharia.** 1 ed. Rio de Janeiro: Elsevier, 2006. 639 p.

ARNOLD, J. R. **Introduction to Materials Management.** Columbus, Ohio: Prentice Hall, 1998. 521 p.

BAKER, K. R. **Introduction to Sequencing and Scheduling.** John Wiley. New York, 1974. 112 p.

BALLOU, R. H. **Gerenciamento da cadeia de suprimentos/logística empresarial.** 5 ed. Porto Alegre: Bookman, 2006. 616p

BARROS, J. R. F. e TUBINO, D. F. **O Planejamento e Controle da Produção nas Pequenas Empresas – Uma Metodologia de Implantação.** 1998. Disponível em <http://www.abepro.org.br/biblioteca/ENEGEP1998_ART262.pdf>. Acesso em setembro de 2017

BLAZEWICZ, J.; ECKER, K. H.; PESCH, E.; SCHIMDT, G.; WEGLARZ, J. **Scheduling Computer and Manufacturing Processes.** Springer – Verlag, Berlin, 1996.

BOWERSOX, D. J.; CLOSS, D. J. **Logística empresarial: o processo de integração da cadeia de suprimento.** São Paulo: Atlas, 2011. 594 p.

CHIAVENATO, I. **Iniciação ao Planejamento e Controle de Produção.** São Paulo: McGraw-Hill, 1990. 116p.

CHIAVENATO, I. **Planejamento e Controle da Produção.** 2. ed. Barueri- SP: Manole, 2008. 159p.

COLIN, E. C. **Pesquisa Operacional: 170 aplicações em Estratégia, Finanças, Logística, Produção, Marketing e vendas.** Rio de Janeiro: LTC, 2013. 501 p.

ERDMANN, Rolf Hermann. **Organização de sistemas de produção.** Florianópolis: Insular, 1998. 216 p.

FAVARETTO, F. **Uma contribuição ao processo de gestão da produção pelo uso da coleta automática de dados de chão de fábrica.** Tese de doutorado. Escola de Engenharia de São Carlos, Universidade de São Paulo, 2001.

FIDELIS, M. B.; ARROYO, J. E. C. **Heurísticas para o problema de sequenciamento de lotes de tarefas em máquinas paralelas.** XLIX Simpósio Brasileiro de Pesquisa Operacional. Blumenau, 2017.

GIL, A. C. **Métodos e técnicas de pesquisa social.** 6 ed. São Paulo: Atlas, 2008.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização Combinatória e Programação Linear: modelos e algoritmos.** 2. ed. Rio de Janeiro: Elsevier, 2005.

GOMES JÚNIOR, A. C. **Problema de Sequenciamento em uma Máquina com Penalidade por Antecipação e Atraso: Modelagem e Resolução.** 86 p. Monografia (Pós-Graduação) – Curso de Engenharia de Produção, Universidade Federal de Minas Gerais, Belo Horizonte, 2007.

GRAVES, S. C. **A Review of Production Scheduling.** *Operations Research*, v29, p. 646-675, 1981.

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional,** 9 ed. Porto Alegre: AMGH, 2013. 1005 p.

HOCHBAUM, D. S. **The Scheduling Problem.** RIOT: Remote Interactive Optimization Testbed, 1999. Disponível em: <<http://riot.ieor.berkeley.edu/Applications/Scheduling/algorithms.html>>. Acesso em: setembro 2017.

JOHNSON, S. M. **Optimal Two- and Three-Stage Production Schedules with Set-Up Times Included.** *Naval Research Logistics Quarterly*, vol 1, 61-68. 1954.

JOO, C. M. e KIM, B. S. **Machine scheduling of time-dependent deteriorating jobs with determining the optimal number of rate modifying activities and the position of the activities.** Journal of Advanced Mechanical Design, Systems, and Manufacturing, 2015. JAMDSM0007– JAMDSM0007.

KÖHLER, V. C. **Algoritmos evolutivos para o problema de sequenciamento de tarefas em máquinas paralelas com tempo de preparação dependentes da sequência.** 94 p. Dissertação (Mestrado) – Programa de Pós-Graduação em Engenharia de Produção. Universidade Federal de Santa Maria, Santa Maria. 2004.

KUO, W.; YANG, D. **Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect.** Taiwan: Elsevier, 2005. Disponível em <https://s3.amazonaws.com/academia.edu.documents/_the_total_completion_time_in.pdf>. Acesso em outubro de 2017.

LE PAPE, C. **Constraint-based scheduling: A tutorial.** Proceedings of the 1st International Summer School on Constraint Programming, 2005.

LEAL, C. A; **Balanceamento de linhas de montagem por meio de modelos de programação linear inteira mista: estudos em indústria automotiva.** 166 p. Trabalho de conclusão de curso – Universidade Tecnológica Federal do Paraná, Curitiba, 2013.

LUH, P. B.; GOU L.; NAGAHORA, T.; TSUJI, M.; YONEDA, K.; HASEGAWA, T.; KYOYA, Y.; KANO, T. **Job Shop Scheduling With Group-Dependent Setups, Finite Buffers, and Long Time Horizon.** Annals of Operations Research, v 76, p: 233-259, 1998.

LUSTOSA, L.; MESQUITA, M.A.; QUELHAS, O.; OLIVEIRA, R. **Planejamento e controle da Produção.** Rio de Janeiro: Elsevier, 2008.

MARINS, F.A.S. **Introdução à pesquisa operacional.** São Paulo: Cultura Acadêmica: Universidade Estadual Paulista, 2011. 176 p.

MONACI, M.; FAÊDA, F. M.; ARROYO, J. E. C.; SANTOS, A. G. **Sequenciamento de tarefas em máquinas paralelas com tempos de preparação e precedência**

entre as tarefas: modelagem e heurísticas construtivas. XLVII SBPO. Pernambuco, 2015.

MOREIRA, D. **Administração da produção e operações.** 2. ed. São Paulo: Pioneira, 1986.

MÜLLER, F. M.; LIMBERGER, S. J. **Uma heurística de trocas para o problema de sequenciamento de tarefas em processadores uniformes.** Journal SciELO Analytics, Rio de Janeiro, 2000. Disponível em <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-74382000000100005>. Acesso em Outubro de 2017.

NEOS GUIDE SERVER, User's Guide to the NEOS Server, Disponível em <<https://neos-guide.org/content/users-guide>>. Acesso em Outubro de 2017.

NOAL, L. C. **Sequenciamento de tarefas em sistemas de pintura balizado por heurísticas do tipo ATC (Apparent Tardiness Cost).** 2015. 72 p. Dissertação (Pós-Graduação) – Curso de Engenharia de Produção, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2015.

OSMAN, I.H.; LAPORTE, G. **Metaheuristics: A bibliography.** Annals of Operations Research, v. 63, p. 511-623, 1996.

PALOMARES, M. **Alternativas de instalação de centros de distribuição de uma empresa multinacional de grande porte utilizando modelagem matemática.** 2016. 100 p. TCC (Graduação) – Curso de Engenharia de Transportes e Logística, Universidade Federal de Santa Catarina, Joinville, 2016.

PINEDO, M.L. **Scheduling - Theory, Algorithms, and Systems.** 3 ed, New York, 2008. Disponível em: <[http://www.math.nsc.ru/LBRT/5/Scheduling/Scheduling_Theory,%20Algorithms,%20and%20Systems\(Pinedo,2008\).pdf](http://www.math.nsc.ru/LBRT/5/Scheduling/Scheduling_Theory,%20Algorithms,%20and%20Systems(Pinedo,2008).pdf)>. Acesso em Setembro de 2017.

PORTER, K.; LITTLE, D.; PECK, M.; ROLLINS, R. **Manufacturing classifications: relationships with production control systems.** Integrated Manufacturing systems, v. 10, no.4, p 189-198, 1999.

RUSSOMANO, Vítor Henrique. **Planejamento e Controle da Produção**. 5. ed. São Paulo: Pioneira, 1995. 320p.

SANTO. L. C. DO E. **Sequenciamento de máquinas através de algoritmos genéticos**. Trabalho de Conclusão de Graduação, Londrina, 2013. Disponível em: <<http://www.uel.br/cce/dc/wp-content/uploads/VersaoPreliminarTCC-LauraEspiritoSanto.pdf>>. Acesso em Setembro de 2017

SANTOS, V. L. A. **Sequenciamento de tarefa em máquinas paralelas com desgastes dependentes da sequência: resolução heurística**. Dissertação (Pós-Graduação) - Curso Ciência da Computação, Universidade Federal de Viçosa, Viçosa, 2016.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. **Administração da produção**. 2. ed. São Paulo: Atlas, 2002.

SPAK, M. D. S. **Proposta de uma metodologia de apoio à tomada de decisão para localização de centros de distribuição no setor varejista de móveis e eletrodomésticos**. 134 f. Dissertação (Mestrado) – Pós Graduação em Engenharia de Produção, Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2012.

TAHA, H. A. **Pesquisa Operacional**. 8. ed. São Paulo: Pearson, 2008. 359 p.

TUBINO, Dalvio Ferrari. **Manual de Planejamento e Controle da Produção**. 2. ed. São Paulo: Atlas, 2007. 215 p.

VOLLMAN, E.T. et al. **Sistemas de Planejamento & Controle da Produção para o gerenciamento da Cadeia de Suprimentos**. 5ª ed. Porto Alegre: Bookman, 2006.

YANG, D.; CHERN, M. **A two-machine flowshop sequencing problem with limited waiting time constraints**. Elsevier, 1995. Disponível em <<http://www.sciencedirect.com/science/article/pii/036083529400026J>>. Acesso em Outubro de 2017

Tabela 8. Prazo e multa para 24 máquinas e 59 tarefas

Tarefa	Prazo	Multa	Tarefa	Prazo	Multa
1	48	0.17	31	120	0.33
2	120	0.33	32	120	0.17
3	72	0.33	33	168	0.33
4	288	0.21	34	72	0.33
5	48	0.33	35	72	0.33
6	72	0.25	36	72	0.17
7	144	0.33	37	72	0.33
8	120	0.33	38	120	0.33
9	96	0.33	39	120	0.33
10	96	0.33	40	120	0.38
11	144	0.33	41	120	0.21
12	192	0.33	42	96	0.33
13	96	0.33	43	96	0.25
14	48	0.33	44	96	0.38
15	120	0.33	45	96	0.38
16	168	0.33	46	72	0.17
17	192	0.33	47	72	0.29
18	48	0.08	48	144	0.08
19	96	0.33	49	72	0.25
20	72	0.33	50	96	0.38
21	144	0.08	51	72	0.33
22	96	0.33	52	72	0.29
23	192	0.33	53	96	0.29
24	72	0.33	54	72	0.33
25	72	0.33	55	72	0.33
26	48	0.33	56	72	0.33
27	312	0.33	57	72	0.38
28	96	0.33	58	72	0.38
29	96	0.33	59	168	0.17
30	120	0.33			

Fonte: Autoria própria