



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE CIÊNCIA E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

AMANDA DA SILVA XAVIER

**SEQUENCIAMENTO PREEMPTIVO E DESIGNAÇÃO DE TAREFAS EM MÁQUINAS
PARALELAS UNIFORMES COM TÉCNICAS DE FLUXO EM REDES**

Recife
2020

AMANDA DA SILVA XAVIER

SEQUENCIAMENTO PREEMPTIVO E DESIGNAÇÃO DE TAREFAS EM MÁQUINAS
PARALELAS UNIFORMES COM TÉCNICAS DE FLUXO EM REDES

Dissertação de Mestrado apresentada ao Departamento de Engenharia de Produção da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Mestre em Engenharia de Produção.

Área de concentração: Pesquisa Operacional

Orientador: Prof. PhD Sóstenes Luiz Soares Lins

Recife
2020

Catálogo na fonte
Bibliotecário Gabriel Luz, CRB-4 / 2222

- X3s Xavier, Amanda da Silva.
Sequenciamento preemptivo e designação de tarefas em máquinas paralelas uniformes com técnicas de fluxo em redes / Amanda da Silva Xavier – Recife, 2020.
84 f.: figs., quads., tabs., siglas e abrev.
- Orientador: Prof. Dr. Sóstenes Luiz Soares Lins.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia de Produção, 2020.
Inclui referências e apêndice.
1. Engenharia de Produção. 2. Sequenciamento de tarefas. 3. Designação de tarefas às máquinas. 4. Placas expandidas em EVA. 5. Fluxo em rede. 6. Problema de Fluxo Máximo. I. Lins, Sóstenes Luiz Soares (Orientador). II. Título.

UFPE

658.5 CDD (22. ed.)

BCTG / 2020-170

AMANDA DA SILVA XAVIER

**SEQUENCIAMENTO PREEMPTIVO E DESIGNAÇÃO DE TAREFAS EM MÁQUINAS
PARALELAS UNIFORMES COM TÉCNICAS DE FLUXO EM REDES E
ABORDAGEM HEURÍSTICA**

Dissertação de Mestrado apresentada ao Departamento de Engenharia de Produção da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Mestre em Engenharia de Produção.

Aprovada em: 28/02/2020.

BANCA EXAMINADORA

Prof. PhD Sóstenes Luiz Soares Lins (Orientador)
Universidade Federal de Pernambuco

Prof. Dr. Isis Didier Lins (Examinador Interno)
Universidade Federal de Pernambuco

Profª. PhD Nivan Roberto Pereira Júnior (Examinador Exteno)
Universidade Federal de Pernambuco

AGRADECIMENTO

Diante desses dois anos de tantos desafios e trabalho, mas também de muito conhecimento adquirido e grandes aprendizados, ainda tenho muito a agradecer.

Agradeço primeiramente a Deus pelas oportunidades que me foram dadas, por ter conhecido pessoas, lugares interessantes e pelas fases difíceis, que foram matérias-primas para o meu aprendizado pessoal e profissional. Por soprar em meus ouvidos “vai ficar tudo bem filha” quando tudo parecia desabar, porque o que realmente importa é o esforço dedicado diariamente. Obrigada Pai.

Tenho muito que agradecer a minha família, aqueles que aceitaram esta minha jornada fora de casa, longe de seus olhos, de seus abraços e de seus conselhos e que me deram o maior apoio do mundo: a confiança de que estou trilhando o caminho certo e entendem que o meu afastamento foi importante para o objetivo maior. Minha mãe Eliane Silva, meu pai Airton Xavier, meus irmãos Alana, Aline, Elielton e Emily Nayara, meus cunhados Pedro Paulo e Thaynara Alencar, minha tia Rosivânia Maria, minha vó Maria Francisca e demais parentes que tanto se orgulham do caminho que venho trilhando. Amo vocês.

Ao meu amado Andersson Alves, que sempre estive ao meu lado nos momentos de alegrias e tristezas, que tolerou minhas crises de ansiedade e me fez melhorar muito como pessoa e como profissional por meio de seus conselhos e broncas. Sou grata por cada crítica feita por ele em meus trabalhos acadêmicos e minha vida pessoal. Gostaria de permanecer junto a você para trilhar mais caminhos potencializadores da vida profissional, pois juntos tornamos as dificuldades em brincadeiras e tudo fica mais leve.

Quero agradecer também ao meu orientador, professor Sóstenes Luiz Soares Lins, que foi como um segundo pai para mim, durante esses dois anos. Graças ao seu bom humor, paciência e atenção o trabalho tornou-se mais leve, além de me ajudar a descobrir um potencial em mim que eu nem conhecia. Foi um prazer imenso trabalhar com um dos maiores matemáticos brasileiros. Te tenho como inspiração professor, obrigada!

Agradeço aos professores e professoras do Programa de Pós-Graduação em Engenharia de Produção (PPGEP) da UFPE pela atenção e conhecimento transmitido por meio de uma didática que facilitou muito meu aprendizado. Em especial, agradeço aos professores: Isis Didier Lins, Márcio José das Chagas Moura, Adiel Teixeira de Almeida Filho, Denise Dumke de Medeiros e

Danielle Costa Moraes. E a Tereza, secretária do PPGEF-UFPE, por sempre oferecer o seu melhor com toda paciência, atenção e organização em todo tipo de problemas e dificuldades surgiram durante esses dois anos.

Agradecer aos meus e minhas colegas do PPGEF-UFPE 2018, a união da turma foi crucial para o aprendizado e para desenvolvimento de trabalhos nas disciplinas. O compartilhamento de conhecimentos com pessoas tão incríveis foi gratificante. Em especial: Emerson Rodrigues, Jerusa, Rafael Veloso, Nádia Cuambe, Júlia, Jéssica, Monaliza, Felipe e Daniel.

Um agradecimento especial à Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) por todo o apoio e financiamento durante todo o mestrado, sem este fomento seria impossível a realização de todos os trabalhos desenvolvidos.

Agradeço aos professores participantes da minha banca de defesa no Mestrado Acadêmico: Isis Didier Lins e Nivan Roberto Ferreira Júnior, por todas as contribuições e sugestões dadas para a melhoria contínua do trabalho.

Ao Centro de Informática (CIn) da UFPE por me receber bem em seus laboratórios e instalações pelos dois anos de mestrado. Meu contato com o CIn instigou uma paixão pela Ciência da Computação e certamente será a minha melhor casa nos próximos 4 anos de Doutorado.

Agradeço aos funcionários da empresa fabricante de EVA (Espuma Vinilica Acetinada) de Barbalha-CE, pelo interesse na pesquisa e por disponibilizar seus dados para validação do programa desenvolvido nesta dissertação.

Ao mais, a todos que não foram citados e contribuíram de forma direta ou indireta para a conclusão deste trabalho.

RESUMO

Estudos voltados para sequenciamento de tarefas, auxiliam os planejadores de produção e vendas ao tomar decisões rápidas enquanto negociam datas de entrega dos pedidos. Por vezes, as técnicas de sequenciamento não são praticadas nas empresas por descaso da análise das consequências que estas podem gerar. Enquanto que na literatura, grande parte dos estudos utilizam técnicas de sequenciamento, mas sem utilizar dados reais. Diante disto, esta pesquisa propõe um método de resolução de problema de sequenciamento preemptivo de tarefas em ambientes de máquinas paralelas uniformes e um algoritmo para designação das tarefas às máquinas. A validação do modelo proposto foi feita com dados reais de uma empresa fabricante de EVA (Espuma Vinílica Acetinada), localizado na cidade de Barbalha-CE. Esta empresa estava com problemas excessivos de atraso na entrega de pedidos da produção de placas expandidas em EVA, oriundas de um mau sequenciamento e falha na designação de tarefas às máquinas. Foi desenvolvido um programa na linguagem de programação *Python*, junto ao *SoPlex*. O programa recebe os dados da empresa em uma planilha de *Excel*, converte em um problema de fluxo em rede, descreve matematicamente a rede em um problema de fluxo máximo, envia o problema para o *solver SoPlex*, converte a resposta do *solver* em tarefas sequenciadas, e designa as tarefas nas máquinas utilizando o algoritmo de designação. O sequenciamento proposto, livre de atrasos, apresentou uma antecipação máxima de aproximadamente 16 dias da data devida. Enquanto que o sequenciamento com a regra FIFO (First In, First Out), utilizado pela empresa, apresentou atraso máximo de 2 dias, representado em 9% do total de tarefas programadas. Foi verificado que com cinco máquinas a empresa consegue atender todas as tarefas conservando o sequenciamento sem atrasos, mas em um tempo de conclusão maior do que no caso de nove máquinas. As tarefas sequenciadas com preempção deixam as máquinas menos sobrecarregadas. Do ponto de vista computacional, o programa é atrativo por consumir menos de 1 segundo no tempo de resposta requerido. Em ambiente de nove máquinas, as regras de priorização (SPT, LPT, LIFO e EDD) são analisadas. A regra LIFO (Last In, First Out) apresentou ganhos analisados em três aspectos. Com 5 máquinas, no ambiente produtivo estudado, as técnicas utilizadas nesta pesquisa são bem mais atrativas que as regras de priorização.

Palavras-chave: Sequenciamento de tarefas. Designação de tarefas às máquinas. Placas expandidas em EVA. Fluxo em rede. Problema de Fluxo Máximo.

ABSTRACT

Task sequencing studies assist production and sales planners in making quick decisions while negotiating order delivery dates. Frequently, the companies are not use practices of the sequencing due to the analysis absence of the consequences that they can generate. While the literature use sequencing techniques, but without using real data. In view of this, this research proposes a method for solving preemptive task sequencing problems in uniform parallel machine and an algorithm for assigning tasks to machines. The validation of the proposed model was done with real data from a company that makes EVA (Ethylene Vinyl Acetate), located in the city of Barbalha-CE. This company was problems delay in the delivery of orders for the production of expanded EVA plaque, because to poor sequencing and failure to assign tasks to the machines. It was designed a program in Python programming language, together with solver SoPlex. The program receives the company's data in an Excel spreadsheet, converts it into a network flow problem, mathematically describes the network into a maximum flow problem, sends the problem to the SoPlex solver, converts the solver's response into sequenced tasks, and assign tasks on machines using the assignment algorithm. The proposed sequencing, without delays, presented approximately 16 days in advance most of the due date. While that in the sequencing with the prioritization rule FIFO (First In, First Out), that the company adopts, the delay most was 2 days represented in 9% of the total scheduled tasks. It was found that with five machines the company could complete all tasks maintaining the sequencing without delays, but in a longer completion time than in the case of 9 machines. Preemptively sequenced tasks leave machines less overloaded. From a computational point of view, the program is attractive because it consumes less than 1 second in the required response time. In a nine machines environment, was analyzed the prioritization rules (SPT, LPT, LIFO e EDD). The LIFO (Last In, First Out) rule highlights gains analyzed in three aspects. With five machines, in the studied productive environment, the techniques used in this research are much more attractive than prioritization rules.

Keywords: Task sequencing. Assigning tasks to machines. Expanded EVA plaque. Network flow. Maximum flow problem.

LISTA DE FIGURAS

Figura 1 - Complexidade dos métodos de resolução de problemas de otimização combinatória .	24
Figura 2-Representação de grafo direcionado (a) e grafo não-direcionado (b).....	31
Figura 3-Representação de um grafo bipartido	31
Figura 4- Representação da rede bipartida utilizada por Martel	36
Figura 5-Representação da rede tripartida utilizada por Federgruen e Groenevelt	37
Figura 6-Representação da rede bipartida utilizada em Xavier, Lins e Silva.....	38
Figura 7 - Etapas de execução da metodologia desta pesquisa	41
Figura 8 – Grafo bipartido para sequenciamento preemptivo de tarefas em máquinas paralelas .	46
Figura 9 - Rede bipartida para sequenciamento preemptivo de tarefas em máquinas paralelas ...	47
Figura 10 - Rede resultante para o sequenciamento	50
Figura 11 - Algoritmo de designação de tarefas às máquinas	52
Figura 12-Comportamento do Lateness no sequenciamento da empresa e proposto	59
Figura 13-Análise com regras de prioridades	64
Figura 14-Análise entre sequenciamento proposto e LIFO (5 máquinas).....	68

LISTA DE QUADROS

Quadro 1 - Definição das regras de prioridades populares.....	22
Quadro 2 - Hipóteses para o problema de sequenciamento de tarefas	43
Quadro 3 – Resumo da avaliação comparativa quanto aos aspectos.....	70

LISTA DE TABELAS

Tabela 1 - Dados de um problema de sequenciamento	44
Tabela 2 – Montagem dos intervalos.....	45
Tabela 3 – Exemplo de designação de tarefas às máquinas	53
Tabela 4 – Exemplo resultante do sequenciamento de tarefas	54
Tabela 5- Dados de entrada para implementação do modelo	57
Tabela 6-Sequenciamento e designação de máquinas da empresa.....	58
Tabela 7-Sequenciamento e designação de máquinas proposto	58
Tabela 8- Detalhamento de dados do L_j	61
Tabela 9-Análise de sensibilidade (alterando o número de máquinas)	62
Tabela 10-Análise de sensibilidade (Designação de tarefas em 5 máquinas)	63
Tabela 11-Sequenciamento e designação da regra SPT	65
Tabela 12- Sequenciamento e designação da regra LPT.....	66
Tabela 13- Sequenciamento e designação da regra LIFO	67
Tabela 14- Sequenciamento e designação da regra LIFO para 5 máquinas	68
Tabela 15- Sequenciamento e designação da regra EDD.....	69

LISTA DE SIGLAS E ABREVIACES

A	Conjunto de arcos de um grafo
<i>EDD</i>	<i>Early Due Date</i>
EVA	Espuma Vinlica Acetinada
<i>FIFO</i>	<i>First In First Out</i>
G	Grafo
<i>LPT</i>	<i>Longest Process Time</i>
N	Conjuntos de ns de um grafo
<i>NP</i>	<i>NonDeterministic polynomial time</i>
<i>NP</i>	<i>CompleteNonDeterministic polynomial time Complete</i>
OPs	Ordens de Produo
OSs	Ordens de Servio
<i>P</i>	<i>Polynomial time</i>
PCP	Planejamento e Controle da Produo
PIM	Programao Inteira Mista
PL	Programao Linear
PPCP	Planejamento Programao e Controle da Produo
<i>SPT</i>	<i>Shortest Processing Time</i>

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	JUSTIFICATIVA	15
1.2	OBJETIVOS	17
1.2.1	Objetivo Geral.....	17
1.2.2	Objetivos Específicos	18
1.3	ESTRUTURA DA DISSERTAÇÃO	18
2	REFERENCIAL TEÓRICO	20
2.1	SISTEMA DE PRODUÇÃO.....	20
2.2	PLANEJAMENTO, PROGRAMAÇÃO E CONTROLE DA PRODUÇÃO (PPCP)	21
2.3	SEQUENCIAMENTO DE TAREFAS	22
2.3.1	Otimização Combinatória e Programação Linear (PL).....	23
2.3.2	Notação e classificação de sequenciamento	25
2.4	TEORIA DOS GRAFOS, FLUXO EM REDES E PROBLEMA DE FLUXO MÁXIMO	30
3	REVISÃO DA LITERATURA.....	33
3.1	MÉTODOS DE SOLUÇÃO.....	33
3.2	PROBLEMAS DE SEQUENCIAMENTO EM FLUXO DE REDE	35
4	METODOLOGIA.....	40
4.1	CLASSIFICAÇÃO DA PESQUISA	40
4.2	DELINEAMENTO DA PESQUISA	40
4.3	APLICABILIDADE DO MODELO USADO NESTA PESQUISA	43
4.4	DEFINIÇÃO DO PROBLEMA	43
4.5	MODELAGEM MATEMÁTICA	44
4.6	<i>PYTHON</i> EM INTERFACE COM <i>SOPLEX</i>	51
5	APLICAÇÃO PRÁTICA.....	55
5.1	COLETA DE DADOS	55
5.2	IMPLEMENTAÇÃO DO MODELO.....	56
5.3	RESULTADOS DO ESTUDO COM DADOS REAIS	57
5.4	SEQUENCIAMENTO PROPOSTO X REGRAS DE PRIORIDADES.....	64

6	CONSIDERAÇÕES FINAIS.....	71
	REFERÊNCIAS.....	74
	APÊNDICE A – AMBIENTE DE 5 MÁQUINAS ($LmÁx$).....	82
	APÊNDICE B –PRINCIPAIS REGRAS DE PRIORIDADES ($LmÁx$).....	83
	APÊNDICE C –AMBIENTE 5 MÁQUINAS PARA REGRA LIFO ($LmÁx$).....	84

1 INTRODUÇÃO

O sequenciamento e designação de tarefas são técnicas cruciais no tocante à competitividade de mercado. Isso leva à busca constante das empresas por atendimento de prazos de entregas firmados com os clientes (PINEDO, 2016). Um *déficit* no uso destas técnicas pode gerar excessos de recursos, que requerem: mais mão de obra, elevados custos, longos processamentos, entre outros. Estes problemas refletem em atrasos, insatisfação do cliente e perda de valor da imagem da empresa (VIEGAS, 2016). Um controle efetivo originado na ordem de processamento das atividades produtivas, pode proporcionar uma resposta rápida e assertiva do sistema de produção (MOREIRA, 2011; SILVA et al., 2018).

O sequenciamento de tarefas é uma técnica voltada para melhorias de processos, originados na carência por controle e monitoramento de sistemas de processamento em lote, seja na manufatura ou no serviço (DONADONI; PEREIRA, 2017). Esta técnica ordena a execução das tarefas, considerando os recursos disponíveis e restrições do processo, a fim de minimizar, por folhaexemplo, o tempo de conclusão das tarefas, tempo de estoques intermediários ou o número de tarefas atrasadas (PINEDO, 2016; TAVARES, 2015). Enquanto que a designação de tarefas busca cumprir o início e fim do processamento de tarefas por meio da melhor alocação das mesmas (MOREIRA, 2011), assim, estas técnicas devem estar atreladas uma a outra.

Na literatura, os problemas de sequenciamento podem ser classificados como: sequenciamento de máquina única, paralela, *flow-shop*, *job-shop* e *open shop*. O sequenciamento de máquinas paralelas é um dos problemas mais importantes, porque considera tanto ambientes de manufatura quanto de serviços (LI et al., 2019). Pinedo (2016), ressalta a importância de estudar ambientes de máquinas paralelas tanto pelos aspectos matemáticos quanto pelos aspectos aplicados. Ele destaca a importância teórica, por tratar de uma generalização de máquina única e um caso especial de *flow-shop*, e a importância prática, devido à ocorrência de recursos em paralelo ser comum no mundo real. Por este motivo, é o ambiente de interesse desta pesquisa.

O ambiente de máquinas paralelas é convencionalmente organizado em três tipos: idênticas, uniformes ou não-relacionadas. Estes três tipos definem se o tempo de processamento vai depender da tarefa (idêntica), da taxa de velocidade da máquina (uniforme) ou da tarefa atribuída à máquina (não relacionadas) (SOPER; STRUSEVICH, 2019). Optou-se por estudar o tipo uniforme, por considerar que mesmo que a empresa tenha comprado uma nova linha de produção (geralmente

mais rápida) e mantenha a antiga linha de produção (mais lenta) em uso, ainda assim é possível sequenciar tarefas considerando as semelhanças entre as máquinas. A taxa de velocidade de processamento afeta o desempenho da programação e os custos de fabricação (KAYVANFAR; ZANDIEH; TEYMOURIAN, 2017).

O problema de designação de tarefas em ambientes de máquinas paralelas é considerado NP-completo por se tratar de um problema combinatório, que requer a alocação de um conjunto de tarefas em um determinado conjunto de máquinas em que cada máquina pode processar no máximo uma tarefa por vez, e cada tarefa pode ser processada por, no máximo, uma máquina por vez (AHUJA; MAGNANTI; ORLIN; 1993, CHERRI et al., 2020).

Diante disto, esta dissertação buscou propor uma solução ao problema de sequenciamento e designação de tarefas em máquinas paralelas uniformes, utilizando dados de uma empresa, localizada no Ceará, que atua na fabricação de EVA (Espuma Vinílica Acetinada). Esta empresa sofre com problemas excessivos de atraso na entrega de pedidos da produção de placas expandidas em EVA e desequilíbrio de cargas nas máquinas. Estes problemas são oriundos de um mau sequenciamento e falha na designação de tarefas às máquinas. As consequências a estes problemas implicam em cancelamentos de pedidos e reprocessamento de cargas (reprogramação).

Para resolver problemas de sequenciamento e designação de tarefas em máquinas paralelas uniformes, foi utilizada uma aplicação apresentada por Ahuja, Magnanti e Orlin (1993), que faz uso da modelagem proposta por Federgruen e Groenevelt (1986). Nesta aplicação, trata-se o problema de sequenciamento de tarefas como um problema de fluxo em rede, que é resolvido como um problema de fluxo máximo. A designação das tarefas é proposta nesta pesquisa por um algoritmo desenvolvido para atribuir cada tarefa à máquina seguindo a ordem do sequenciamento encontrado.

1.1 JUSTIFICATIVA

As atividades na fase de programação da produção exigem que sejam minimizados o tempo de produção e custos, considerando as condições do planejamento alinhado com as estratégias da empresa. Estas decisões estão voltadas basicamente para a designação de tarefas às máquinas e a programação das mesmas em cada máquina (ARENALES et al., 2007). Muitas empresas delegam estas atividades aos gerentes de produção, e por vezes, a mesma é repassada aos líderes de equipe,

que programam sem análise das consequências de desempenho global de um mau sequenciamento de tarefas (FUCHIGAMI; RANGEL, 2014). Ou seja, as técnicas de sequenciamento, geralmente, não são praticadas nas empresas para a fase de programação da produção.

Muitos trabalhos encontrados na literatura, relacionados ao problema de sequenciamento em máquinas paralelas uniformes utilizam abordagens heurísticas com simulação de dados, apesar da ampla aplicabilidade citada em Pinedo (2016). Assim as técnicas de sequenciamento são consideradas, mas sem utilizar dados reais e limitando os dados de entrada do modelo proposto.

O trabalho de Lushchacova (2009), Koulamas e Kyparisis (2009), por exemplo, resolvem o problema considerando apenas duas máquinas paralelas uniformes. No primeiro trabalho é utilizado uma variante da regra *SPT* (*Shortest Processing Time*) com tarefas disponíveis simultaneamente. Os autores sugerem tratar os tempos de processamentos comuns entre as tarefas e as datas de início conhecidas. No segundo trabalho, é utilizado um algoritmo *LPT* (*Longest Process Time*) modificado. O problema é tratado sem restrição de processamento ou de início, mas com tempos de processamentos diferentes.

O trabalho de Soper e Strusevich (2019) mostra que os problemas de máquinas paralelas uniformes restringidos a uma única preempção é solucionado mais rápido quando considerado até duas máquinas. Preempção significa que o processamento de uma tarefa pode ser interrompido, e continuando mais tarde em qualquer máquina. Nenhum destes trabalhos utilizaram abordagens exatas para resoluções deste tipo de problema, apenas métodos heurísticos.

Em relação a designação de tarefas grande parte dos trabalhos (por exemplo: Zolfaghari, El-Bouri e Namiranian (2007), McDonald et al. (2009) ou Bard e Wan (2006)) utilizam modelos que necessitam de muitas variáveis tornando o problema NP-difícil, por trata-se de um problema combinatório. Do ponto de vista prático em sistema produtivo real, principalmente de pequenas empresas, nem sempre é possível obter todos os dados históricos necessários referentes a essas variáveis.

Nesse contexto é justificável o desenvolvimento de um programa que sequencie e designe as tarefas às máquinas em tempo hábil para sistemas de produção real, que atue com máquinas paralelas uniformes e permita preempção no seu processo produtivo. Assim, o uso de uma modelagem exata para o problema de sequenciamento de tarefas e um algoritmo para o problema da designação de tarefas, que mantém a solução exata encontrada no sequenciamento de tarefas,

tornará hábil o tempo de execução do programa. Para avaliar o tempo de execução do programa em sistema produtivo real, é interessante o uso de dados reais.

Uma empresa fabricante de EVA concordou em disponibilizar seus dados para esta pesquisa. Assim o programa utilizará dados como número de máquinas, tempos de processamentos, início e fim de tarefas conhecidos e fixados (cedidos pelos responsáveis pelo processo produtivo). A finalidade é propor uma solução para o problema de sequenciamento de tarefas e designação de tarefas com uso de um programa prático.

No processo de fabricação das placas de EVA, um *déficit* no sequenciamento de tarefas, ou seja, uma falha no modo de priorização das Ordens de Produção (OPs ou tarefas), bem como a má designação destas às máquinas afetam diretamente a produtividade, uma vez que: *i*) aumentam o tempo de conclusão das tarefas, *ii*) prejudicam o fluxo de produção do processo com aumento de estoque intermediário, *iii*) atrasam os pedidos comprometendo os ativos da empresa, e sua credibilidade no mercado.

Deste modo, esta pesquisa contribui financeiramente tanto para a empresa, com um meio de sequenciar e designar suas OPs às máquinas reduzindo cancelamento de pedidos por atrasos, quanto aos clientes, por melhorar o nível de atendimento de prazos. No meio acadêmico, esta pesquisa contribui com uso de uma modelagem exata sem restringir: número de máquinas, interrupções no processo (preempção), tempos de processamento, tempos de início e fim. Esta pesquisa estende os estudos Xavier, Lins e Silva (2019) automatizando as etapas de aplicação apresentadas no trabalho e acrescentando a etapa de designação das tarefas às máquinas.

1.2 OBJETIVOS

Nesta seção são apresentados o objetivo geral que contempla essa dissertação, bem como seus objetivos específicos, que descrevem os passos executados no delineamento da pesquisa.

1.2.1 Objetivo Geral

Proposição de um método: modelagem do problema de sequenciamento de tarefas em ambientes de máquinas paralelas como um problema de fluxo máximo considerando preempção e

desenvolvimento de um algoritmo para designação das tarefas às máquinas preservando o sequenciamento sugerido.

1.2.2 Objetivos Específicos

- a) Estudar trabalhos da literatura que tratam o problema abordado e relacionados;
- b) Verificar a aplicabilidade do modelo matemático para o sequenciamento de tarefas;
- c) Desenvolver um algoritmo para o problema da designação das tarefas nas máquinas;
- d) Desenvolver um programa utilizando linguagem de programação *Python 3.7* para ser executado junto ao *solver SoPlex* para aplicação do modelo matemático e do algoritmo;
- e) Validar o programa com dados de um sistema produtivo real e comparar os resultados obtidos com o cenário atual do mesmo;
- f) Fazer análise de sensibilidade quanto ao número de máquinas para identificar possíveis melhorias para o sistema produtivo; e,
- g) Avaliar o comportamento do sequenciamento proposto com as principais regras de prioridades comumente discutidas na literatura.

1.3 ESTRUTURA DA DISSERTAÇÃO

Os demais capítulos desta dissertação têm seus conteúdos resumidos a seguir:

Capítulo 2. Este capítulo aborda os aspectos de sistema de produção com foco no PPCP, sequenciamento de tarefas e problemas de fluxo máximo. Verifica-se a literatura associada ao problema objeto tratado nesta dissertação.

Capítulo 3. Neste capítulo é apresentada uma revisão da literatura sobre os estudos que abordam os problemas de sequenciamento de máquinas paralelas uniformes em que se diferenciam seus métodos de solução bem como objetivos. São apresentados, também, os poucos trabalhos presentes na literatura que reduzem estes problemas a problemas de fluxo em rede.

Capítulo 4. Este capítulo apresenta o método de solução utilizado nesta pesquisa para o problema do *Maximum Utilization*, no sequenciamento de tarefas em ambientes de máquinas paralelas uniformes. Neste método inclui-se o uso de um modelo de Programação Linear originado

de uma rede bipartida, um algoritmo de designação de tarefas e os softwares utilizados para viabilização da aplicação do modelo em um sistema produtivo real.

Capítulo 5. Neste capítulo é apresentado a validação do método proposto nesta pesquisa, a partir de dados de um sistema produtivo real. Os principais objetivos deste capítulo são: ilustrar a aplicação e os resultados obtidos na utilização do modelo descrito com o sequenciamento do sistema produtivo, descrevendo os resultados segundo o procedimento utilizado no delineamento deste estudo; e, comparar o resultado do sequenciamento proposto com as principais regras de prioridade encontradas na literatura.

Capítulo 6. Este capítulo finaliza o estudo com as considerações finais, onde é enfatizado o alcance dos objetivos pretendidos. São destacadas, também, as limitações e dificuldades enfrentadas nesta pesquisa e recomendadas sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo aborda os aspectos de sistema de produção com foco no PPCP, sequenciamento de tarefas e problemas de fluxo máximo. A seção 2.1 trata da caracterização do sistema de produção voltado para importância da presença de um PPCP, o qual é discutido na seção 2.3. A seção 2.4 trata da definição, classificação e notação de sequenciamento de tarefas em seus diferentes ambientes e regras de priorização. O capítulo finaliza com algumas notações e definições oriundas da teoria dos grafos (seção 2.5).

2.1 SISTEMA DE PRODUÇÃO

Um Sistema de Produção é composto por um conjunto de partes inter-relacionadas que transformam *inputs* (entradas) em *outputs* (saída). Os *inputs* referem-se a todo tipo de insumos que deverão entrar em um dado processo, com recursos capazes de causar algum tipo de transformação para que seja convertido em algo novo, denominado, *output*. Em síntese, pode-se afirmar que Sistema de Produção é a parte da empresa que transforma a matéria-prima em produto acabado, intermediando fornecedor, produção e cliente final (SILVA; ESTENDER; BARBOSA, 2015; ESTENDER et al., 2017).

Neste contexto, é preciso que os sistemas produtivos sejam pensados em termos de prazos, em que ações planejadas são executadas na transformação dos *inputs* em *outputs*. Assim, o planejamento pode ser dividido em três níveis: de longo prazo (Plano de Produção), de médio prazo (Plano Mestre) e de curto prazo (Programação) (TUBINO, 2007). Estes níveis, quando bem definidos no ambiente organizacional, mantêm a empresa competitiva diante de seus concorrentes e indiretamente auxiliam em questões cruciais ao ambiente organizacional, funcionando como um instrumento de detecção de ações necessárias demandadas por gestores (SILVA; ESTENDER; BARBOSA, 2015).

No Plano de Produção (estratégico), é feita a previsão de vendas e verificação do atendimento do pedido do cliente de acordo com a capacidade produtiva. O Plano Mestre (tático), busca táticas para operar com efetividade da capacidade instalada, a fim de atender os pedidos em carteira já negociados (TUBINO, 2007). A Programação (operacional), é responsável por operar o sistema produtivo dentro da tática montada no Plano Mestre. É neste nível, que são gerenciados os estoques,

sequenciadas as ordens de produção, emitidas, liberadas as ordens de compras e executado o acompanhamento e controle da produção (TUBINO, 2007; SCHÄFERS et al., 2019; LUSTOSA et al., 2008).

De acordo com Tubino (2007), ter em mãos estas atividades designadas nos níveis de tempo, significa que a empresa agora deve buscar montar um setor ou departamento de apoio a produção em muitos casos denominados PCP (Planejamento e Controle da Produção) ou PPCP (Planejamento, Programação e Controle da Produção), o qual atenderá ao planejamento estabelecido nos três níveis, da melhor maneira possível.

2.2 PLANEJAMENTO, PROGRAMAÇÃO E CONTROLE DA PRODUÇÃO (PPCP)

O PPCP tem o objetivo de apoiar e coordenar o sistema de produção para chegar a um determinado resultado (SANTOS; SANTOS, 2018). Este setor recebe informações de outros setores, tais como: compras, vendas, finanças, almoxarifados, contabilidade, recursos humanos entre outros; processa as informações recebidas e distribui em outros setores, caracterizando o setor de PPCP como um sistema que delega operações produtivas, pois são destas informações que este consegue realizar o planejamento da produção (ALBUQUERQUE et al., 2017).

O planejamento é a seleção do próximo passo de ação na produção ou o plano com especificações de ações a serem realizadas em determinado período (HITOMI, 2017). As informações obtidas no planejamento geram OPs e subsequência à programação, onde são definidas quando os produtos serão fabricados atribuindo os prazos nos quais as tarefas deverão ser executadas (GUERRINI; BELHOT; AZZOLINNI JÚNIOR, 2018).

Deste modo, o controle atua na missão de medir e corrigir o desempenho das atividades executadas em função do planejamento e programação, a fim de assegurar que os objetivos estejam sendo cumpridos (SLACK; CHAMBERS; JOHNSTON, 2009). Estes objetivos podem ser resumidos em: minimizar a ociosidade dos recursos produtivos, reduzir estoques, atendimentos de ordem de produção, bem como minimizar *leadtimes* da produção (OLIVEIRA, 2017).

O PPCP é, portanto, uma área crucial para o alcance de metas e atendimento de prazos de entregas firmados com os clientes. Estes mantem a união entre seu sistema produtivo e as estratégias da empresa (curto, médio ou longo prazo) (OLIVEIRA, 2017) sem deixar de operar

com diversos recursos de dados. Além disso, o PPCP monitora e controla o desempenho do sistema produtivo em relação ao que foi planejado e programado (GONÇALVES, 2018).

O PPCP formaliza o que se espera que aconteça, pois, um planejamento não garante que o que foi programado ocorra como esperado (TUBINO, 2007) e por vezes, os tempos previstos na programação não são atendidos. Os motivos para isto, se devem às eventualidades não planejadas, tais como: ocupação de máquinas com tarefas de maior prioridade, manutenção, quebra ou até mesmo por atraso de fornecedor (PINEDO, 2016; GUERRINI; BELHOT; AZZOLINI JÚNIOR, 2018). Logo, se há o risco de não atendimento dos prazos e espera de tarefas a serem processadas, então há necessidade de determinar a ordem de realizações das mesmas para um efetivo controle. Essa fase é nomeada de sequenciamento de tarefas (MOREIRA, 2011).

2.3 SEQUENCIAMENTO DE TAREFAS

Sequenciamento de tarefas é uma técnica inserida no setor de PPCP, capaz de auxiliar os planejadores de produção e vendas ao tomar decisões rápidas enquanto negociam, com seus clientes, as datas de entrega (SILVA; FUCHIGAMI, 2018). Os objetivos desta técnica incluem minimização do tempo de conclusão das tarefas, dos tempos de estoques intermediários ou do número de tarefas atrasadas, impondo regras de prioridades, definidas no Quadro 1, que considerem os recursos disponíveis e restrições de processo (PINEDO, 2016; TAVARES, 2015).

Quadro 1 - Definição das regras de prioridades populares

Regra de prioridade	Descrição
<i>SPT (Shortest Processing Time)</i>	Prioriza o pedido com menor tempo de processamento
<i>LPT (Longest Processing Time)</i>	Prioriza o pedido com maior tempo de processamento
<i>FIFO (First In First Out)</i>	Prioriza o pedido pela ordem de chegada
<i>LIFO (Last In Last Out)</i>	Prioriza o pedido pela última OP a chegar no sistema
<i>EDD (Early Due Date)</i>	Prioriza o pedido pela menor data de entrega

Fonte: Adaptado de Pinedo (2012) e Fuchigami e Silva (2018)

As regras de prioridades são voltadas para a seleção da tarefa que terá prioridade de processamento, a partir das ordens de produção e/ou das condições do sistema produtivo (TUBINO, 2007). A variação no tempo de chegadas das tarefas, classificam estas regras em dinâmicas (tarefas chegam em tempos diferentes) ou estáticas (tarefas chegam simultaneamente) (MOREIRA, 2011; SILVA; FUCHIGAMI, 2018). As regras estáticas permitem uma visão completa no sistema produtivo, de modo que o sequenciamento busque encontrar uma tarefa adequada para um determinado recurso. Enquanto que, nas regras dinâmicas, o sequenciamento faz a mesma busca toda vez que uma nova tarefa é inserida no sistema (MENDIS; AUDSLEY; INDRUSIAK, 2017).

Os métodos de sequenciamento apresentados no Quadro 1 são, consideravelmente, mais fáceis de implementar, mas permitem a obtenção de soluções razoáveis. Em problemas mais simples podem até alcançar uma boa solução, por exemplo: a regra SPT garante menor tempo de fluxo total e EDD menor atraso máximo. Porém, quando se trata de um problema mais complexo, as regras de prioridades só serão úteis se incorporadas a meta-heurísticas, as quais não garantem o ótimo (ABREU; PEREIRA, 2018).

O ótimo é garantido por meio dos métodos exatos, no entanto, o tempo gasto por estes métodos para encontrar uma solução ótima para um problema NP-difícil, se esse método existe, é de uma ordem de magnitude muito maior que o da heurística. Sendo, em muitos casos, inaplicável (MARTÍ; REINELT, 2011).

Segundo Közen e Minner (2017), é possível obter um desempenho quase ótimo empregando regras simples. Por exemplo: sempre que um custo fixo de atraso está envolvido, a política de sequenciamento de prioridade ideal desvia do princípio da regra EDD. No entanto, uma regra de EDD ajustada tem um bom desempenho. Além disso, o adiamento das decisões de sequenciamento de prioridade até a próxima conclusão melhora o desempenho. Em Pesquisa Operacional, o sequenciamento de tarefas em uma ou várias máquinas é feito utilizando técnicas de otimização combinatória (YALAOUI et al., 2010).

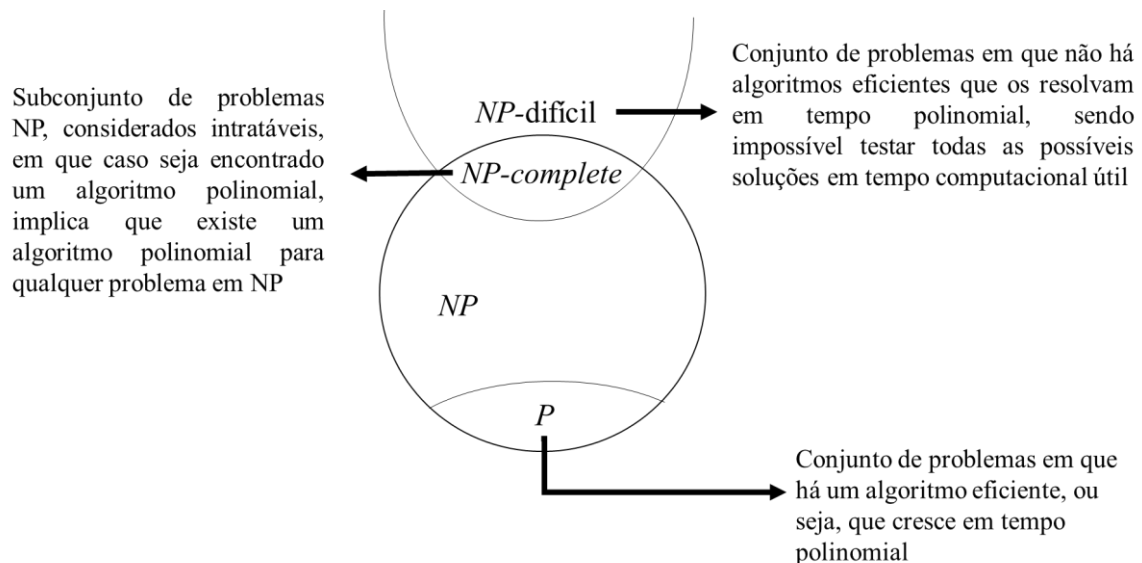
2.3.1 Otimização Combinatória e Programação Linear (PL)

A Otimização Combinatória é a área da Pesquisa Operacional que busca um meio eficiente de projetar e operar um determinado sistema. De modo que as decisões sejam tomadas com base

na melhor combinação de valores para as variáveis do problema, atendendo às restrições de processamento, em tempo de execução computacional considerável útil (COTA, 2018).

Segundo Schrijver (2003), a classificação da eficiência do método de resolução de problemas de otimização combinatória, com base no tempo de execução computacional, ocorreu entre os anos 1960 e 1970. Iniciando com Edmonds, em 1960, quando defendeu a ideia de classificar o método como eficiente se o seu tempo de execução estiver limitado por um polinômio (P). Enquanto que em 1970, Cook e Karp descobriram que vários outros problemas de otimização combinatória são mais difíceis, criando, assim, uma nova classe denominada tempo polinomial Não Determinístico (NP). Perante isto, é possível dividir a eficiência dos métodos em três tipos, como é visto na Figura 1.

Figura 1 - Complexidade dos métodos de resolução de problemas de otimização combinatória



Fonte: Adaptado de Schrijver (2003) e Veríssimo (2016)

Como representado na Figura 1, a classe NP inclui a maioria dos problemas de otimização combinatória. Qualquer problema em NP pode ser reduzido aos problemas NP -Complete. Todo problema de NP -complete é equivalente no sentido de que a resolução em tempo polinomial de um deles implica o mesmo para todos (GRÖTSCHEL; LOVÁSZ; SCHRIJVER, 2012).

Problemas de otimização não são problemas de decisão. Problemas de decisão requerem que sejam tomadas decisões do tipo sim ou não. Os problemas de decisão podem ser associados aos problemas de otimização utilizando Programação Linear. Os Problemas de Programação Linear (PPL) assumem uma maximização e/ou minimização de uma função de muitas variáveis, sujeita

às restrições de desigualdade, igualdade e/ou restrições de integralidade em algumas ou todas as variáveis. Se houver um algoritmo de tempo polinomial que resolva o problema de otimização, então o problema de decisão associado é resolvido de forma eficiente (GRÖTSCHEL; LOVÁSZ; SCHRIJVER, 2012; WOLSEY; NEMHAUSER, 2014).

O PPL consiste em uma técnica matemática para resolver problemas de otimização linear (PARIS, 2016). O termo “Linear”, significa que todas as funções envolvidas (Função Objetivo e restrições) são lineares. A palavra “Programação” atua como sinônimo de planejamento. Assim, a “Programação Linear” planeja atividades para obter um resultado ótimo entre todas as alternativas viáveis (HILLIER; LIEBERMAN, 2013).

Problemas sequenciamento de tarefas em sistemas com várias máquinas, são tratados como problemas combinatórios complexos do tipo *NP*-difícil (XIAO et al., 2016). Um meio, encontrado pelos pesquisadores, de resolvê-los em tempo polinomial é formulando-o como um PPL, que é polinomial. Para isto, é necessário conhecer todas as características que envolvem o sequenciamento estudado, tais como: o ambiente de máquina, as restrições de processamento e o objetivo a ser minimizado, como é descrito na próxima seção.

2.3.2 Notação e classificação de sequenciamento

Na literatura, os problemas de sequenciamento geralmente são descritos pelas tripla: $\alpha|\beta|\gamma$. Em que o α representa o ambiente de máquina, o qual contém somente uma entrada. O β detalha características e restrições de processamento, podendo ter uma ou mais entradas, bem como nenhuma entrada. O γ descreve o objetivo a ser minimizado, usualmente contém apenas uma entrada (PINEDO, 2012; ALLAHVERDI, 2015).

Os ambientes de máquinas (α) tratam da quantidade de recursos disponíveis, bem como o comportamento do fluxo de processamento das OPs/OSs. Dentre os mais citados na literatura, destacam-se:

- a) Ambiente de máquina única trata-se de um problema de otimização que sequencia um conjunto de tarefas em apenas uma máquina (ALLAHVERDI, 2015; NIU et al., 2019);
- b) Ambiente de máquinas paralelas trata-se de uma generalização do ambiente de máquina única, onde há um conjunto de recursos ou máquinas capazes de processarem quaisquer atividades ou tarefas (MUNDIM; FUCHIGAMI, 2017). Estas podem ser classificadas

como: idênticas, uniformes ou não relacionadas, definidas a seguir por Pinedo (2012), Allahverdi (2015) e Habiba et al. (2018);

- Para o caso de máquinas paralelas idênticas, existem m máquinas em paralelo com velocidades iguais para uma mesma tarefa, ou seja, o tempo de processamento P_j depende apenas da tarefa j . A tarefa j requer uma única operação e pode ser processada em qualquer uma das m máquinas ou em qualquer uma que pertença a um determinado subconjunto de máquinas M_j . Este ambiente é representado por P_m no campo α ;
 - Em máquinas paralelas uniformes, existem m máquinas dispostas em paralelo com velocidades diferentes, mas com as mesmas características, considerando que o tempo de processamento das tarefas dependem apenas do tempo que a máquina gasta para as processar. A velocidade da máquina i é indicada por uma taxa de velocidade v_i . O tempo que a tarefa j passa na máquina i (P_{ij}) é igual a $\left(\frac{P_j}{v_i}\right)$ (supondo que a tarefa j receba todo o processamento da máquina i). Este ambiente é representado por Q_m no campo α . Se todas as máquinas tiverem a mesma velocidade, ou seja, $v_i = 1$ para todos os i e $P_{ij} = P_j$, o ambiente será idêntico ao anterior;
 - No ambiente de máquinas paralelas não relacionadas, o tempo de processamento depende tanto da tarefa quanto da máquina. Assim, a máquina i pode processar a tarefa j na velocidade v_{ij} . O tempo de processamento P_{ij} que a máquina i gasta para processar a tarefa j será $\left(\frac{P_j}{v_{ij}}\right)$ (novamente assumindo que a tarefa j receba todo o processamento da máquina i). Este ambiente é representado por R_m no campo α ;
- c) Em ambiente *Flow Shop* (fluxo de operações) existem n tarefas a serem processadas em m máquinas em série, que seguem uma mesma sequência (todas as tarefas seguem a mesma rota), sendo que uma tarefa não pode ser entregue a próxima máquina antes que todo o processamento na máquina corrente seja concluído. Depois de concluído o processamento, a tarefa junta-se à fila da próxima máquina (MENG et al., 2018; HOSSEINABADI et al., 2019). Este ambiente é representado por F_m no campo α (ALLAHVERDI, 2015);

- d) Em ambientes de máquina *Job shop* (intermitente) existem n tarefas que devem ser processadas por m máquinas. Cada tarefa é formada por uma sequência de operações consecutivas e cada operação requer exatamente uma máquina. A decisão refere-se à sequência das operações nas máquinas (NOUIRI et al., 2018). Cada tarefa tem uma rota específica (HOSSEINABADI et al., 2019). Este ambiente é representado por J_m no campo α (ALLAHVERDI, 2015);
- e) Em ambiente *Open Shop* (ambiente aberto) são considerados n tarefas a serem processadas por m máquinas, em que cada tarefa j consiste em operações independentes O_{ij} . No sequenciamento programado é atribuído P_{ij} unidades de tempo para cada operação O_{ij} , de modo a assegurar que nenhum trabalho seja processado simultaneamente em duas máquinas diferentes e que nenhuma máquina processe simultaneamente dois trabalhos diferentes (WEGINGER, 2018). Neste caso, não há restrições quanto ao roteamento de cada tarefa através do ambiente da máquina (HOSSEINABADI et al., 2019). Este ambiente é representado por O_m no campo α (ALLAHVERDI, 2015).

Entre estes ambientes de máquinas apresentados, o sequenciamento de máquinas paralelas é um dos problemas mais importantes, devido a sua ampla aplicação que vai além do ambiente fabril (LI et al., 2019). Na literatura, estudos que utilizaram este ambiente em sistema de informação, por exemplo, consideraram prestadores de serviços paralelos: *call centers* e operações bancárias (BALIN, 2011). Enquanto que este mesmo ambiente em uma indústria aeronáutica, por exemplo, as operações de montagem de um avião podem ser consideradas atividades em paralelo (SILVA; MORABITO; YANASSE, 2011).

As características e restrições de processamento (β) especificam, nos problemas de sequenciamento, as limitações no sistema de produção estudado. Este é um campo que pode incluir múltiplas entradas, dentre as quais citam-se: Preempção (*prmp*), Restrição de Precedência (*prec*), Início de processamento (r_j), restrição de elegibilidade de máquina (M_j), entre outras. Estas estão definidas, a seguir, por Pinedo (2012).

- a) A característica de Preempção (*prmp*) significa que o processamento de uma tarefa pode ser interrompido e continuado mais tarde em qualquer máquina. Vale ressaltar que a quantidade de processamento que a tarefa interrompida já recebeu não é perdida. A tarefa

interrompida é colocada em outra máquina, no caso de máquinas paralelas, e só precisa da máquina pelo tempo restante de processamento;

- b) No caso da Restrição de Precedência (*prec*), a próxima tarefa só pode ser iniciada depois que a tarefa anterior concluir. Se cada tarefa tiver no máximo um sucessor, as restrições são chamadas de *intree*. Se cada tarefa tiver no máximo um predecessor, as restrições são chamadas de *outtree*. Se cada tarefa tiver no máximo um predecessor e no máximo um sucessor, esta restrição é referida com *chain*;
- c) O início de processamento (r_j) aparece no campo β para indicar que a tarefa não pode ser iniciada antes do período especificado neste campo;
- d) A restrição de elegibilidade de máquina (M_j) aparece apenas quando o ambiente de máquina é o de máquinas paralelas idênticas (P_m). Esta restrição de processamento indica a quantidade de máquinas que podem processar a tarefa j , não necessariamente indica que a mesma irá passar em todo o conjunto de máquinas M_j , mas em um subconjunto de máquinas ($M_j \subset M$).

Os objetivos a serem minimizados (γ) podem estar em função do tempo de conclusão C_{ij} da tarefa j na máquina i , que depende do sequenciamento proposto (o tempo de conclusão na última máquina a qual requer processamento é referenciado por C_j). Ou em função da data prometida ao cliente (d_j) (PINEDO, 2012), no qual é analisado o cálculo feito pelos atrasos nomeados: *Lateness*, *Tardiness* e *Unit Penalty*, definidos a seguir por Pinedo (2012) e Zarei et al. (2016).

- a) *Lateness* (L_j): É a diferença algébrica entre a data devida (d_j) e o real tempo de conclusão (C_j) de uma tarefa (equação 1). O resultado negativo indica que a tarefa foi concluída antes de d_j . Por outro lado, o resultado positivo indica que a tarefa foi concluída depois de d_j . O resultado zero indica que não houve atraso:

$$L_j = C_j - d_j \quad (1)$$

- b) *Tardiness* (T_j): Considera apenas a diferença positiva no mesmo cálculo do *Lateness*, indicando quantas unidades de tempo a tarefa passou de d_j para ser concluída (equação 2);

$$T_j = \max(C_j - d_j, 0) \quad (2)$$

- c) *Unit Penalty* (U_j): Calcula a penalidade unitária para a quantidade de tarefas atrasadas. Neste caso, em cada tarefa é verificado se o real tempo de conclusão C_j da tarefa passou de d_j , como indicado na equação 3.

$$U_j = \begin{cases} 1, & \text{se } C_j > d_j \\ 0, & \text{caso contrário} \end{cases} \quad (3)$$

Em relação aos objetivos de minimização voltados para o tempo de conclusão C_j citam-se quatro principais critérios de otimização: Atraso máximo (*Maximum Lateness*), Tempo total de conclusão ponderado (*Total weighted completion time*), *Makespan* e Utilização máxima (*Maximum Utilization*).

O *Maximum Lateness* ($L_{\max}$) busca minimizar maior atraso de entrega (ABREU; PEREIRA, 2018). O *Total weighted completion time* ($\sum w_j C_j$), busca minimizar o custo de programação da produção representada pela soma ponderada dos tempos de conclusão das tarefas (SITTERS; YANG, 2018). O *Makespan* ($C_{\max}$), busca minimizar o instante de tempo de conclusão da última tarefa a deixar o sistema. Quando os tempos de *setup* não dependem da sequência o *Makespan* é igual a $\sum P_{ij}$, que representa a soma dos tempos de processamento das tarefas nas máquinas (PINEDO, 2012).

O *Maximum utilization* implica em boa utilização dos recursos (máquinas) e redução do desequilíbrio de cargas nas máquinas (PINEDO 2012; HABIBA et al., 2018). A *Maximum utilization* está vinculado à minimização do *Makespan* (BENDER; RABIN, 2002; ZHOU et al., 2016), entretanto, minimizar *Makespan* não equivale a maximizar a utilização dos recursos (*Maximum Utilization*).

Federgruen e Groenevelt (1986) mostram como os critérios de otimização podem ser otimizados por algoritmos de caminho de aumento quando aplicados a uma rede escolhida adequadamente. No referido problema *Maximum Utilization*, um sequenciamento viável é encontrado a partir de uma determinada coleção de intervalos de tempos disjuntos, em busca de minimizar a quantidade máxima de tarefas atribuídas a cada um desses intervalos. Este critério é particularmente relevante quando se tenta criar uma quantidade máxima de capacidade de folga em cada um dos intervalos de tempos considerados (FEDERGRUEN; GROENEVELT, 1986; ALCOFORADO; BORNSTEIN, 2002). Isso implica na redução do tempo máximo de conclusão das tarefas sem atrasos (ALCOFORADO; BORNSTEIN, 2002).

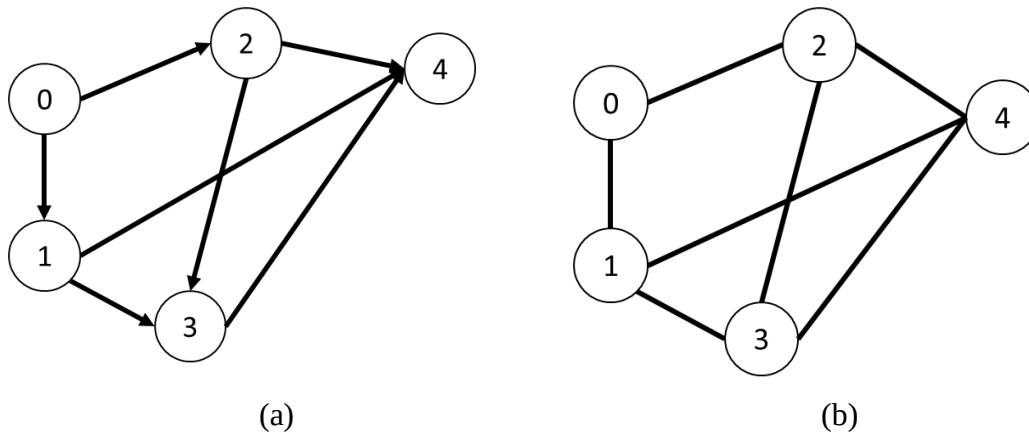
O primeiro algoritmo polinomial elaborado para o problema *Maximum utilization* mais geral, foi construído por Charles Martel em 1982. O autor mostra que um sequenciamento viável pode ser determinado resolvendo um problema de fluxo em rede polimatroidal. O fluxo máximo pode ser encontrado aplicando técnicas análogas àquelas para um problema de fluxo em rede padrão. Para abordar o problema de fluxo máximo, algumas notações e definições oriundas da teoria dos grafos precisam ser definidas.

2.4 TEORIA DOS GRAFOS, FLUXO EM REDES E PROBLEMA DE FLUXO MÁXIMO

A Teoria dos Grafos é o campo da matemática que estuda grafos (estruturas capazes de modelar diversos problemas, por meio de representação matemática de objetos chamados vértices e se há relação entre eles, arestas/arcos). O primeiro problema, em Teoria dos Grafos, foi modelado por Leonhard Euler, em 1736, com o *problema das pontes*, em que buscava-se descobrir sobre a possibilidade de fazer um passeio pela cidade Königsberg, na Alemanha, começando e terminando no mesmo lugar, passando exatamente uma vez por cada uma das 7 pontes, que cruzam o rio Pregel. Euler chegou à conclusão de que não é possível (PRESTES, 2016; NETTO; JURKIEWICZ, 2017).

Um grafo é definido por $G = (N, A)$, em que N se refere a um conjunto não vazio de nós/vértices e A ao conjunto de arcos (ligações direcionadas) e/ou arestas (ligações não-direcionadas) que interligam os nós. Um grafo pode ser direcionado (digrafo), Figura 2a, quando estas ligações forem arcos; ou não-direcionado, Figura 2b., quando as ligações entre os vértices forem arestas. Em cada arco ou aresta é atribuída uma capacidade não negativa, $c_{ij} > 0$, que limita o quanto de fluxo pode ser transmitido do nó inicial ao nó final (KELLER; TROTTER, 2015). Uma rede é um grafo com capacidades nos arcos e/ou nas arestas.

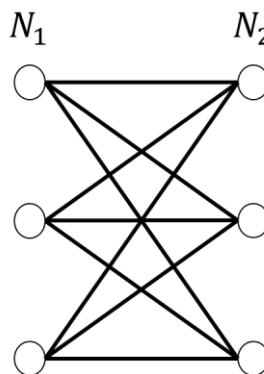
Figura 2-Representação de grafo direcionado (a) e grafo não-direcionado (b)



Fonte: Adaptado de Prestes (2016)

Segundo Prestes (2016), um tipo especial de grafo é chamado grafo k -partido, se: i) o conjunto de nós puder ser particionado em k conjuntos não vazios $N_1, N_2, \dots, N_k$, de modo que sejam disjuntos dois a dois, ou seja, $N_i \cap N_j = \emptyset$ para todo $i \neq j$, e a união destes elementos formam um único conjunto, ou seja, $N_i \cup N_j = N$; ii) cada arco $a \in A$ tem suas extremidades em conjuntos distintos, ou seja, se $a = \{u, v\}$, então $u \in N_i$ e $v \in N_j$, onde $i \neq j$. Um grafo 2-partido é um caso particular chamado de grafo bipartido e está exemplificado na Figura 3.

Figura 3-Representação de um grafo bipartido



Fonte: Adaptado de Prestes (2016)

O problema de fluxo máximo considera uma rede onde é desejável enviar o máximo de fluxo possível entre os conjuntos de nós (N) e arcos (A), iniciando em um nó fonte (s) e finalizando em um nó sorvedouro (t), sem exceder a capacidade c_{ij} de qualquer arco, e considerando a

conservação de fluxo (KELLER; TROTTER, 2015; TUCKER, 2012). A cada par de arco (i, j) é associado um limite inferior no fluxo de $l_{ij} = 0$ e um limite superior no fluxo de $l_{ij} = c_{ij}$ (BAZARAA; JARVIS; SHERALI, 2010). Seja f a quantidade de fluxo na rede entre o nó s e o nó t , seja x_{ij} o fluxo entre i e j , ou seja, nos arcos (i, j) . Matematicamente pode-se representar o problema de fluxo máximo a seguir:

$$\text{Maximize } f \quad (4)$$

sujeito a:

$$\sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(j,i) \in A\}} x_{ji} = \begin{cases} f & \text{se } i = s \\ 0 & \forall i \in N - \{s, t\} \\ -f & \text{se } i = t \end{cases} \quad (5)$$

$$0 \leq x_{ij} \leq c_{ij} \quad \forall (i, j) \in A \quad (6)$$

Em (4) busca-se maximizar o objetivo de encontrar o fluxo máximo. A restrição (5) garante que haverá conservação do fluxo nos nós, ou seja, todo o fluxo recebido do nó anterior deve ser enviado ao nó seguinte. Vale ressaltar que o nó s e o nó t servem apenas para direcionarem o fluxo, sendo assim s apenas envia fluxo, enquanto t apenas recebe. A restrição (6) limita as capacidades dos arcos e garante a não negatividade dos mesmos.

De acordo com Ahuja, Magnanti e Orlin (1993), a solução para os problemas de fluxo máximo pode ser encontrada dentro de dois grupos: algoritmos de caminho de aumento e algoritmos de *Preflow-push*. O primeiro algoritmo incrementa fluxos ao longo do caminho $s - t$ respeitando a conservação do fluxo nos nós, enquanto que o segundo atua empurrando o fluxo da origem, gerando excessos nos nós, que são enviados para o sorvedouro (t) ou para o nó inicial (s).

Para o grupo de algoritmos de caminho de aumento, a otimalidade é encontrada quando no fluxo do grafo não há mais caminhos de aumento da origem para o sorvedouro. A formalização desta, pode ser observada em um algoritmo, comumente, utilizado para resolver problema de fluxo máximo por meio de caminhos de aumento, o algoritmo de Ford-Fulkerson. Este algoritmo é empregado em problemas de fluxo em rede em que é desejado encontrar um fluxo de valor máximo que respeite as capacidades dos arcos, fazendo o melhor uso destas na rede estudada (SCHRIJVER, 2017).

3 REVISÃO DA LITERATURA

Neste capítulo é apresentada uma revisão da literatura sobre os estudos que abordam os problemas de sequenciamento de máquinas paralelas uniformes em que se diferenciam seus métodos de solução bem como objetivos (seção 3.1). A seção 3.2 retrata os poucos problemas presentes na literatura que reduzem estes problemas a fluxo em rede.

3.1 MÉTODOS DE SOLUÇÃO

A teoria do sequenciamento tem sido estudada, na área de Pesquisa Operacional, desde 1950 (KAHRAMAN; YAVUZ et al., 2010). Muitas destes estudos são voltados para o ambiente de máquinas paralelas uniformes, diferenciando-se nos métodos de soluções, restrições de processamento e objetivos do sequenciamento. Segundo Lee, Leung e Pinedo (2010), os objetivos mais comuns na literatura incluem: *Makespan*, tempo total ponderado de conclusão, número de tarefas atrasadas, atraso máximo ponderado e atraso total ponderado.

Os autores Otten, Braaksma e Boucherie (2019) consideram o problema de sequenciamento estocástico e usaram a regra *Shortest Variance First (SVF)* com objetivo de minimizar o desvio esperado do *Earliness/Tardiness (E/T)*, para resolver o problema de custo e frustrações de pacientes e funcionários relacionados a cirurgias precoces ou tardias em um hospital. Os autores afirmam que o problema de sequenciamento estocástico E/T, minimiza o desvio dos tempos de conclusão das cirurgias, em relação aos tempos de conclusões esperados.

Li, Xiao e Yang (2019) consideram processamento de tarefas em duas máquinas paralelas uniformes uma das máquinas apresenta uma velocidade rápida com custo de processamento mais alto e outra mais lenta com custo de processamento mais baixo. Os tempos de processamento são independentes, as datas devidas das tarefas são conhecidas e diferentes entre si e o processo não permite interrupção (preempção). O objetivo é minimizar o atraso total. Para este problema os autores propõem um modelo de Programação Inteira Mista (PIM) e duas heurísticas com um procedimento de melhoria para obter soluções iniciais. Os resultados são comparados com as soluções exatas obtidas pelo *CPLEX*.

No trabalho desenvolvido por Li et al. (2019), o problema de sequenciamento estudado considera uma única operação em máquinas paralelas uniformes com consumo de recursos por

unidade de tempo ao processar uma tarefa. Assim, o objetivo de minimizar *Makespan* estava sujeito às restrições no consumo total de recursos que não poderiam ultrapassar um determinado valor. Para resolver esse problema, é empregado um algoritmo *Simplified Swarm Optimization (SSO)* e testado, a eficácia, com *Genetic Algorithm (GA)* e um *Particle Swarm Optimization* com *genetic Local Search (PSO_{LS})*.

Em Zhou et al. (2016), é estudado o problema de programação de máquinas paralelas uniformes de processamento em lote de tamanhos arbitrários. Essas máquinas de processamento em lote têm capacidades não idênticas e velocidades diferentes. O objetivo é minimizar *Makespan*. É proposto um algoritmo híbrido baseado em evolução diferencial para solucionar problemas de larga escala. Os indivíduos da população são representados como sequências de trabalho distintas, e novos operadores de mutação e cruzamento são projetados com base nessa representação. Uma heurística, então, é desenvolvida para formar lotes e sequenciá-los nas máquinas paralelas uniformes.

O trabalho desenvolvido por Soper de Strusevich (2019) considera apenas duas máquinas paralelas uniformes, com restrição de uma única preempção e objetivo de minimizar *Makespan*. Os autores afirmam que mais de duas máquinas com esta restrição de preempção, é considerado o pior caso da produção de uma produção ideal. Em Lushchakova (2009), também são consideradas apenas duas máquinas paralelas uniformes, porém, os tempos de processamentos considerados são todos iguais, a preempção é permitida e/ou cada tarefa é conhecida a data de início quando a tarefa fica disponível para processamento. O objetivo é minimizar o tempo total de conclusão.

Kim (2011) considerou minimizar o tempo de conclusão ponderado, sujeito à restrição de precedência, nomeado pelos autores como *s-precedence*, em que a tarefa j é restrita do processamento à tarefa i iniciar, que é diferente da definição padrão de uma relação de precedência onde j não pode iniciar até que i seja concluída. Os autores esclarecem que a regra *s-precedence* ocorre assim: se a tarefa i precede a tarefa j , então a tarefa j não pode iniciar o processamento antes que a tarefa i comece, como uma regra *FIFO*. Um procedimento heurístico baseado em Programação Linear é proposto. São encontradas soluções efetivas e experimentos computacionais são executados.

Esta pesquisa se diferencia dos trabalhos citados acima por utilizar técnicas de fluxo em redes para tratar do problema de sequenciamento de tarefas, de modo que a quantidade de máquinas, tempos de processamento, data início e data devida sejam arbitrários. Além disso, nesta pesquisa

o sequenciamento é resolvido em função de um problema de fluxo máximo modelado em um problema de Programação Linear.

3.2 PROBLEMAS DE SEQUENCIAMENTO EM FLUXO DE REDE

Os casos em que se busca verificar a existência de um sequenciamento viável, desde que os tempos de processamento sejam conhecidos e fixados e a preempção seja permitida, podem ser reduzidos a um problema fluxo em rede. Essa temática é bastante antiga e apresentada nesta seção.

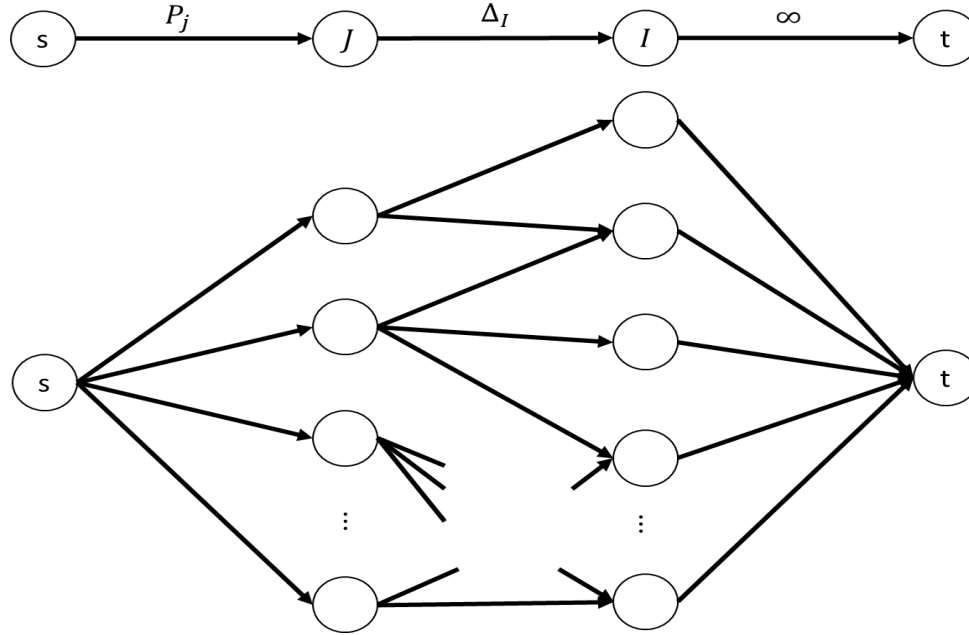
Em Horn (1974), foram considerados ambientes de sequenciamento com um único processador ou um conjunto de máquinas idênticas, em que as tarefas requerem apenas uma operação e interrupções são permitidas. O autor mostra que o problema pode ser convertido para um problema de fluxo de rede.

Gonzalez e Sahni (1978) consideraram tarefas independentes da sequência, com tempos de início e prazos de conclusão comuns entre elas, sendo ordenadas pelo tempo de execução das mesmas, enquanto que os processadores foram considerados uniformes e ordenados pela taxa de velocidade de processamento.

Na contribuição de Sahni e Cho (1979) também foram consideradas tarefas independentes da sequência e processadores uniformes, porém os tempos de liberação das tarefas eram variáveis conservando apenas a ideia do prazo de conclusão comuns. E os autores concluíram que o algoritmo foi eficiente tanto para estas considerações quanto para diferentes tempos de entrega, além de poder ser usado para minimizar o atraso máximo.

Martel (1982) trouxe uma grande contribuição para a literatura com base nestes três trabalhos citados anteriormente. Em vez de ter tempo de liberação e um prazo, ele fez com que cada tarefa recebesse uma lista de intervalos de tempos nos quais as tarefas podem ser processadas. Cada máquina tem também uma lista de intervalos de tempos durante os quais ela pode ser usada. A distribuição destes intervalos na linha do tempo acontece de modo que não haja mudança, sendo então a chave que facilitou a transformação deste problema em fluxo de rede (Figura 4).

Figura 4- Representação da rede bipartida utilizada por Martel



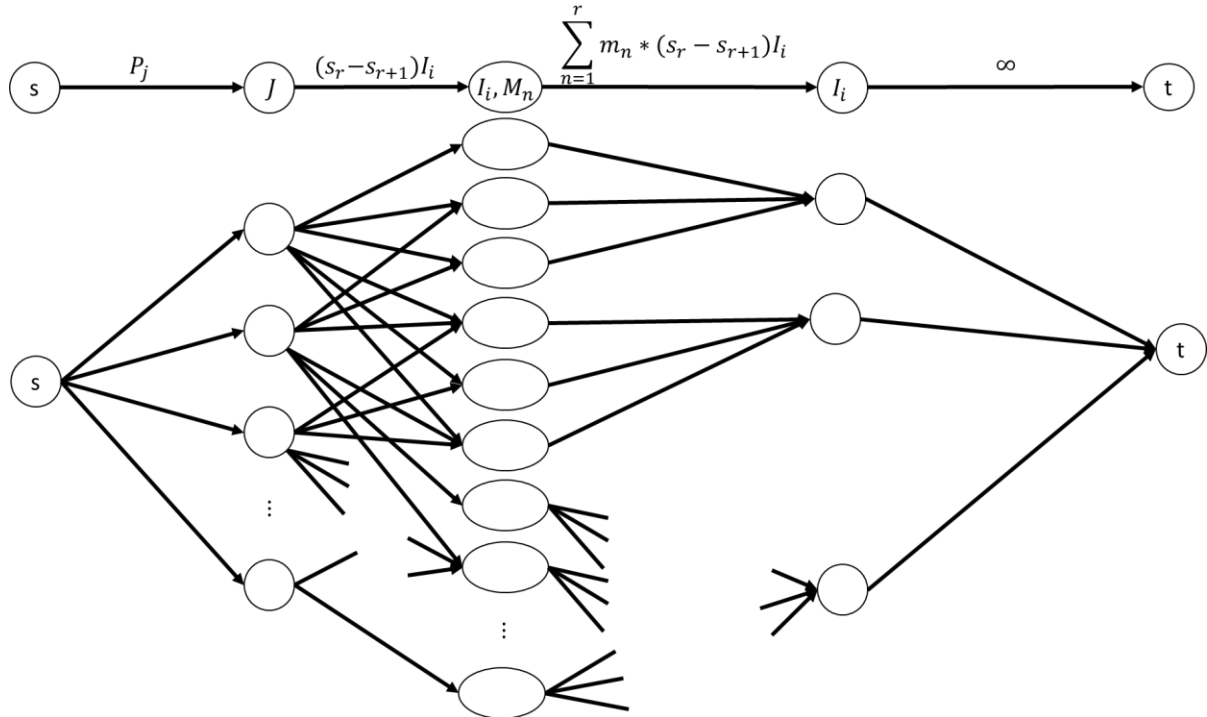
Fonte: Adaptado de Martel (1982)

O grafo definido por Martel (1982) utiliza um grafo bipartido $G = (N, A)$, onde o conjunto N de nós representa o conjunto J das tarefas e I dos intervalos, ou seja, $N = J \cup I$. O autor buscou representar o problema de sequenciamento como um problema de fluxo máximo, em que os arcos de s a j , $j \in J$, tenham capacidade limitada pelo tempo de processamento (considerando as m máquinas) efetivamente alocado à tarefa j ; os arcos de J aos intervalos de I sejam capacitados pelo tempo de processamento da tarefa j efetivamente alocado ao período I , não sendo especificadas as máquinas correspondentes, e os arcos dos intervalos de I a t sejam representados pelo tempo total de processamento alocado ao período $i \in I$.

Federgruen e Groenevelt (1986) expandiram o modelo de fluxo em rede bipartida (Figura 4) sugerido por Martel (1982) para o caso de máquinas uniformes em uma rede tripartida. A rede apresentada por Martel evita um terceiro conjunto de nós $N_3 \in N$ em um grafo $G = (N, A)$. Esse terceiro conjunto, de acordo com Federgruen e Groenevelt (1986), representam os intervalos pertencentes as máquinas, nomeado pelos autores como períodos de máquinas denotados por $I_i M_n$, ou seja, cada intervalo é particionado pela quantidade de máquinas. Isso significa que a rede proposta por Martel permite encontrar apenas o sequenciamento, enquanto que a de Federgruen e

Groenevelt encontra o sequenciamento e designa as tarefas às máquinas simultaneamente, tratando de uma rede mais complexa, como é visto na Figura 5.

Figura 5-Representação da rede tripartida utilizada por Federgruen e Groenevelt



Fonte: Adaptado de Federgruen e Groenevelt (1986) e Shioura, Shakhlevich e Strusevich (2018)

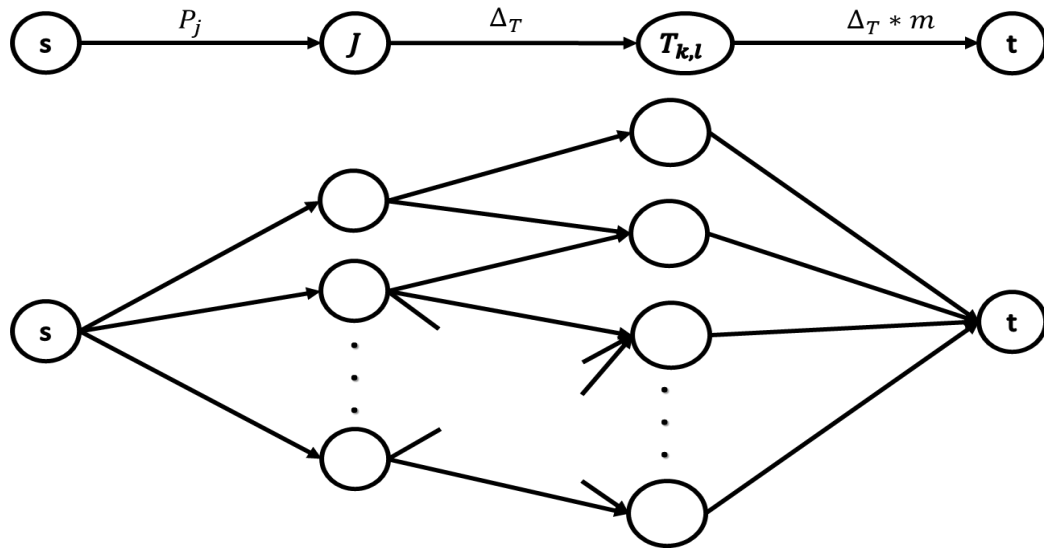
A rede utilizada por Federgruen e Groenevelt (1986) considera um grafo tripartido $G = (N, A)$ onde o conjunto N de nós representa o conjunto J das tarefas, $I_i M_n$ dos períodos correspondentes às máquinas e I_i dos intervalos de tempos, ou seja, $N = J \cup I_i M_n \cup I_i$. A origem s é conectada a cada nó da tarefa j , com capacidade $P_j, j = (1, \dots, n)$, assim como na rede de Martel (1982). Uma tarefa j é conectada a todos os nós do período da máquina $I_i M_n$ em que a tarefa está disponível, correspondente ao tipo de máquina r , com capacidade $(s_r - s_{r+1}) I_i$. Todo nó $I_i M_n$ está conectado ao nó do intervalo correspondente com capacidade $\sum_{n=1}^r m_n * (s_r - s_{r+1}) I_i$. Todo nó do intervalo I_i está conectado ao nó final t do fluxo com capacidade infinita. Semelhante à rede de Martel, o fluxo total de f_{ij} na rede Federgruen e Groenevelt de um nó de tarefa j para um intervalo I_i define a quantidade de processamento da tarefa j no intervalo I_i .

O trabalho de Shakhlevich e Strusevich (2008), trata do problema de máquinas paralelas uniformes, em que as tarefas têm tempos de processamento controláveis e as datas de início e fim arbitrárias ou datas de início em zero e datas devidas comuns. Os autores utilizaram uma

abordagem gulosa para desenvolver um algoritmo que resolva ambos problemas. Este algoritmo utiliza qualquer um dos dois fluxos em redes: Martel (1982) ou Federgroen e Groenevelt (1986).

O trabalho mais recente que reduz o problema de sequenciamento em fluxo de rede, é estudado por Xavier, Lins e Silva (2019). Os autores basearam-se nas redes propostas por Martel (1982), com o método de Federgroen e Groenevelt (1986) a partir de uma aplicação de Ahuja, Magnanti e Orlin (1993). Assim, a rede utilizada pelos autores é representada na Figura 6.

Figura 6-Representação da rede bipartida utilizada em Xavier, Lins e Silva



Fonte: Xavier, Lins e Silva (2019)

A rede utilizada em Xavier, Lins e Silva (2019) utiliza um grafo bipartido $G = (N, A)$, onde o conjunto N de nós representa o conjunto J das tarefas e $T_{k,l}$ dos intervalos, ou seja, $N = J \cup T_{k,l}$. Os autores mostram como tratar o problema de sequenciamento como um problema de fluxo máximo, com o objetivo de maximizar o tempo de processamento como meio de induzir a conclusão de todas as tarefas, por isto os arcos de s a j , $j \in J$, têm capacidade limitada pelo tempo de processamento (considerando as taxas de velocidades v_i das m máquinas) efetivamente alocado à tarefa j ; os arcos entre J e os intervalos de $T_{k,l}$ são capacitados pelo tempo de processamento da tarefa j efetivamente alocado ao intervalo $T_{k,l}$. E os arcos entre os intervalos de $T_{k,l}$ a t são representados pelo tempo total que o intervalo disponibiliza considerando a quantidade de máquinas.

Esta dissertação é uma extensão do trabalho de Xavier, Lins e Silva (2019), com a automatização das etapas de montagem da rede para identificação do sequenciamento de tarefas junto ao algoritmo de designação de tarefas às máquinas sem perder o sequenciamento proposto.

4 METODOLOGIA

Neste capítulo é apresentado um modelo de Programação Linear, originado de uma rede bipartida, que representa um problema de fluxo máximo, com objetivo de resolver o problema de *Maximum Utilization*, no sequenciamento de tarefas em máquinas paralelas uniformes. As tarefas, têm seus tempos de início r_j definidos e permitem interrupção (preempção) em seu processo. Este problema é denotado pela tripla: $(Q_m | r_j, prmp | \sum P_{ij})$.

O capítulo está dividido em: classificação da pesquisa (seção 4.1); delineamento da pesquisa (seção 4.2); aplicabilidade da do modelo (seção 4.3); modelagem matemática (seção 4.4) e uma breve discussão sobre a utilização do *Python* junto ao *SoPlex* no desenvolvimento do programa para este estudo (seção 4.5).

4.1 CLASSIFICAÇÃO DA PESQUISA

Esta dissertação trata de uma pesquisa aplicada, de acordo com sua natureza. Para Gerhardt e Silveira (2009) a pesquisa aplicada tem a finalidade de gerar conhecimentos a partir da aplicação prática, quando explicados problemas específicos com verdades e interesses locais. Quanto aos objetivos, classifica-se como exploratória. Segundo Gil (2010) este tipo de pesquisa proporciona familiarização com o problema para torna-lo mais explícito. Para isto, esta pesquisa fez levantamento bibliográfico, entrevistas com pessoas envolvidas no processo produtivo e análises que estimulam a compreensão.

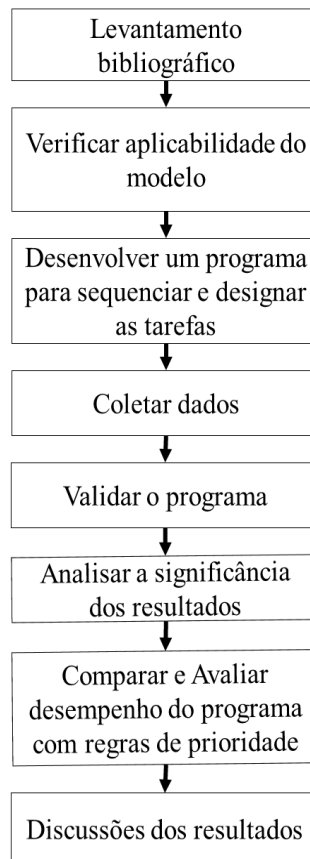
Em relação aos procedimentos, esta pesquisa se encaixa na modalidade de modelagem. De acordo com Ganga (2012) a modelagem descreve o funcionamento de um sistema produtivo ou parte deste, por meio de técnicas matemáticas. Este tipo de pesquisa reúne dados como restrições e variáveis para processar as informações de um problema do cenário em estudo (GANGA, 2012).

4.2 DELINEAMENTO DA PESQUISA

Inicialmente realizou-se um levantamento bibliográfico sobre os principais temas utilizados neste trabalho: sequenciamento de tarefas, problema de fluxo máximo, rede bipartida, *Maximum Utilization*, entre outros relacionados. Esta primeira etapa foi fundamental para obtenção de um

melhor embasamento dos temas abordados e para contribuir com o conhecimento amplo acerca dos assuntos aqui discutidos. Para estruturação de uma sequência lógica na realização do trabalho, elaborou-se um fluxograma (Figura 7) com sete etapas que contemplam este estudo, incluindo a primeira etapa aqui comentada.

Figura 7 - Etapas de execução da metodologia desta pesquisa



Fonte: A autora (2020)

A segunda etapa do delineamento desta pesquisa consistiu em verificar a aplicabilidade do modelo de Ahuja, Magnanti e Orlin (1993). Esta etapa requereu a verificação na literatura de trabalhos que tenham utilizado tal aplicação e a significância dos resultados obtidos com os mesmos.

A terceira etapa referiu-se ao desenvolvimento de um programa que viabilize a aplicação do modelo em sistema produtivo real. Nesta etapa, foi utilizada a linguagem de programação Python e o solucionador de Programação Linear, *SoPlex*. A função do Python seguiu os passos da modelagem matemática (seção 4.5) e está descrito na seção 4.6. O programa faz interface com o

solver SoPlex e converte a solução encontrada em tarefas sequenciadas para então designá-las às máquinas.

A sequência no delineamento da pesquisa, após o desenvolvimento do programa, segue para a quarta etapa, que consiste na coleta de dados. Esta etapa envolve *i*) a atividade de verificar se as características do sistema produtivo atendem às restrições de processamento requisitadas para o tipo de sequenciamento estudado, tais como: preempção, r_j e *ii*) obtenção dos dados de entrada para o modelo, tais como: número de tarefas, tempo de processamento, data devida, taxa de velocidade de máquinas e quantidade de máquinas.

Os dados coletados foram utilizados na etapa cinco, para validação do modelo. Nesta etapa verificou-se a significância dos resultados obtidos para o sistema de produção. Estes resultados foram analisados comparando com o tipo de regra de priorização de tarefas que a empresa adota, caracterizando a sexta etapa.

Na análise dos resultados foi verificado se o sequenciamento proposto apresentou atrasos nas tarefas ou não. É provado em Federgruen e Groenevelt (1986), que o problema de sequenciamento considerado não tem atrasos se o fluxo máximo for igual a $\sum P_j$. Caso contrário, o sequenciamento não apresenta atrasos. Fohler (2011) afirma que para este caso, são possíveis novas tentativas, por exemplo, alterando as propriedades do conjunto de tarefas para uma análise nos dados estudados para verificar o que a empresa poderia melhorar quanto a prazo das tarefas, número de máquinas, entre outros.

Na sétima etapa foram feitas análises de sensibilidade nos dados estudados a partir da variação do número de máquinas para verificar o comportamento do sequenciamento de tarefas e do programa (tempo computacional, $L_{máx}$, $C_{máx}$ e número de interações no *SoPlex*), em busca de possíveis melhorias para o sistema de produção.

A eficiência do modelo foi verificada fazendo a comparação entre os resultados obtidos com o sequenciamento proposto e as regras de prioridade (*FIFO*, *SPT*, *LPT*, *LIFO* e *EDD*). O objetivo desta análise foi verificar o comportamento dos sequenciamentos baseados nos aspectos: atraso máximo ($L_{máx}$), utilização de máquinas, e tempo total de conclusão ($C_{máx}$). O delineamento da pesquisa encerra com as discussões sobre os benefícios obtidos com o programa, compondo a última etapa.

4.3 APLICABILIDADE DO MODELO USADO NESTA PESQUISA

A aplicabilidade do modelo foi verificada na revisão da literatura em que apenas o trabalho de Xavier, Lins e Silva (2019) utilizou a mesma abordagem utilizada por esta pesquisa. Os autores consideram um ambiente de máquinas paralelas uniformes com datas de início, datas devidas, tempo de processamento e número de máquinas conhecidos e fixados, com preempção. No estudo é feita uma aplicação utilizando dados reais cedidos por uma empresa que atua com dois sistemas produtivos (Calçados injetados e EVA).

No trabalho de Xavier, Lins e Silva (2019) foi utilizado um método exato, resolvendo por Programação Linear um problema de fluxo máximo no *software SoPlex* em um caso real com pequenas instâncias. Eles fizeram essa aplicação prática utilizando o modelo de Federgruen e Groenevelt (1986) reduzido a uma rede mais simples encontrada no livro de Ahuja, Magnanti e Orlin (1993), sem descartar a ideia de inserção do número de máquinas. Os autores executam o modelo manualmente e não fizeram a designação de tarefas nas máquinas.

4.4 DEFINIÇÃO DO PROBLEMA

A definição formal deste problema considera: um conjunto J de tarefas em m máquinas paralelas uniformes, em que cada $j \in J$ tem um requisito de tempo de processamento P_j para concluir uma tarefa j , um tempo de início de processamento nomeado r_j e uma data de entrega prometida ao cliente denotada por d_j . As características deste problema são descritas em 4 hipóteses como segue no Quadro 2.

Quadro 2 - Hipóteses para o problema de sequenciamento de tarefas

H1	Cada máquina pode processar no máximo uma tarefa por vez, e cada tarefa pode ser processada por, no máximo, uma máquina por vez;
H2	O número de tarefas e máquinas devem ser conhecidos e fixados;
H3	Interrupções (preempções) são permitidas, isto é, o processamento de uma tarefa pode ser interrompido, continuando mais tarde em qualquer máquina;
H4	r_j e d_j são valores inteiros e positivos;

Fonte: Adaptado de Ahuja, Magnanti e Orlin (1993)

O sistema produtivo e os dados coletados para utilizar o modelo apresentado por esta pesquisa devem estar dentro das restrições de processamento descritos nestas quatro hipóteses. Assumindo as quatro hipóteses segue-se para a modelagem matemática, a qual representa a formulação do problema de sequenciamento em um problema de fluxo máximo modelado em um grafo bipartido.

4.5 MODELAGEM MATEMÁTICA

Nesta pesquisa, o problema de sequenciamento é tratado como um problema de fluxo máximo em uma rede bipartida $G = (N, A)$, onde $N = N_1 \cup N_2$. O conjunto de nós N_1 representam as n tarefas; o conjunto de nós N_2 representam os intervalos de tempos, denominados $T_{k,l}$ (k é limite inferior e l é limite superior do intervalo); e o conjunto A representa os arcos que conectam os nós. É introduzido um nó s (origem) e um nó t (destino) auxiliares, apenas para demonstrar onde deve iniciar e finalizar o fluxo na rede.

A formulação dos intervalos $T_{k,l}$ seguem 3 passos: i) a união dos conjuntos de tempos de início e fim das tarefas formam um único conjunto T com $s_1, s_2, \dots, s_T$ elementos; ii) os elementos do conjunto T são ranqueados em ordem crescente de modo que cada $s \in T$ não se repita; iii) o primeiro elemento da lista (s_i) representa o primeiro limite inferior k do intervalo, enquanto que o segundo elemento sofre uma subtração ($s_{i+1} - 1$), para representar o limite superior l do intervalo $T_{k,l}$ e assim sucessivamente para todos os elementos do conjunto T . Um exemplo é mostrado na Tabela 1 e na Tabela 2.

Tabela 1 - Dados de um problema de sequenciamento

Tarefa (j)	1	2	3	4
Data início (r_j)	1	3	5	3
Data devida (d_j)	8	12	11	10
Processamento (P_j)	4	2,5	3,8	5

Fonte: A autora (2020)

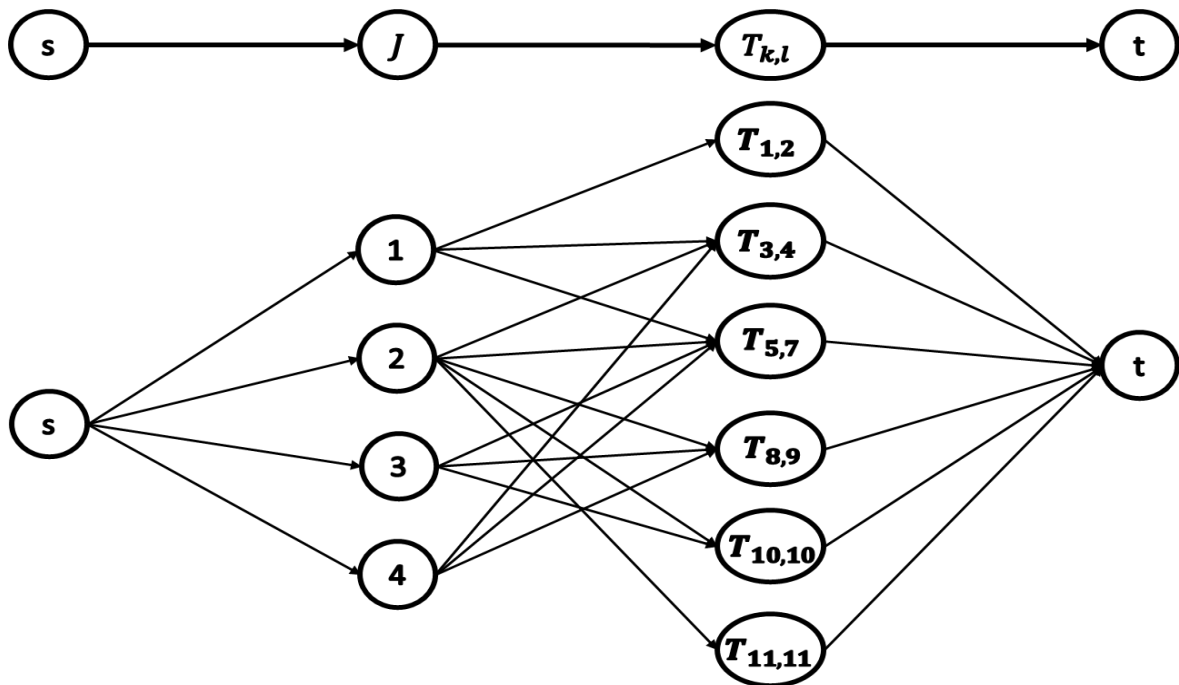
Tabela 2 – Montagem dos intervalos

i)	1, 3, 5, 3, 8, 12, 11, 10
ii)	1, 3, 5, 8, 10, 11, 12
iii)	$T_{k_1 l_1} = T_{1,2};$ $T_{k_2 l_2} = T_{3,4};$ $T_{k_3 l_3} = T_{5,7};$ $T_{k_4 l_4} = T_{8,9};$ $T_{k_5 l_5} = T_{10,10};$ $T_{k_6 l_6} = T_{11,11}.$

Fonte: A autora (2020)

Definidos os intervalos $T_{k,l}$, segue-se para a montagem do grafo $G = (N, A)$ (Figura 8), ainda com os dados do exemplo da Tabela 1 e Tabela 2. O nó s é conectado a cada tarefa j ; cada nó do intervalo $T_{k,l}$ é conectado ao nó t . Enquanto que os arcos entre as tarefas e os intervalos são conectados, se o tempo de início (r_j) da tarefa j for no mínimo o limite inferior k do intervalo $T_{k,l}$ ($r_j \geq k$) e, se, o tempo de entrega $d_j - 1$ for no máximo o limite superior l do intervalo $T_{k,l}$ ($d_j - 1 \leq l$). A regra para conexão entre as tarefas e os intervalos é considerada a chave para eliminação de tarefas que naturalmente estariam atrasadas, pois induz a tarefa a ser finalizada antes da data prevista considerando $d_j - 1$ como limite superior.

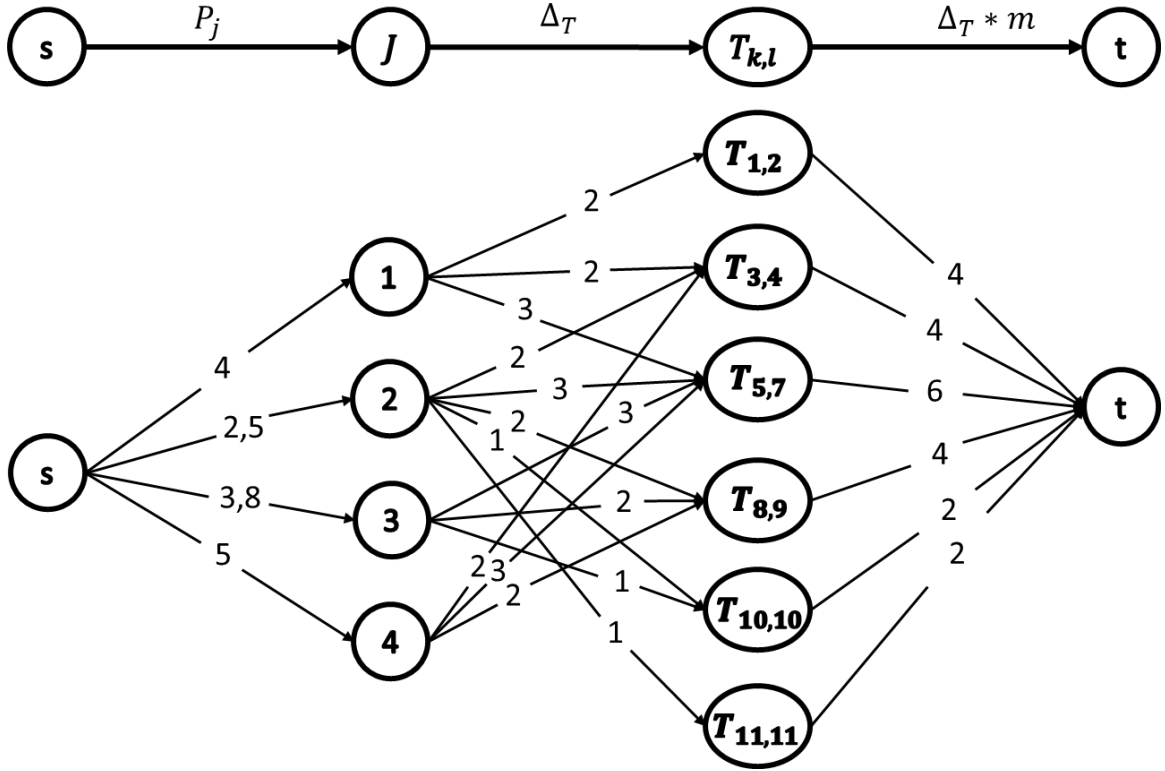
Figura 8 – Grafo bipartido para sequenciamento preemptivo de tarefas em máquinas paralelas



Fonte: A autora (2020)

Em relação as capacidades correspondentes aos os arcos, o nó s e as tarefas j são conectadas por um arco de capacidade P_j , indicando o tempo de máquina necessário para concluir a tarefa j ; os arcos entre as tarefas e os intervalos são capacitados por $\Delta_T = ((l + 1) - k)$, que representa o tempo máximo que o intervalo disponibiliza para alocar um conjunto de tarefas j efetivamente introduzido ao período $T_{k,l}$. Por fim, cada nó do intervalo $T_{k,l}$ é conectado ao nó t , com capacidade $(\Delta_T * m)$ que representa o tempo máximo de disponibilidade de máquina, pela relação do tempo que o intervalo disponibiliza com a quantidade de máquinas disponíveis não sendo especificada a qual máquina pertence cada intervalo. Utilizando o exemplo do grafo G da Figura 8 e considerando um total de 2 máquinas como recurso disponível, construiu-se a rede apresentada na Figura 9.

Figura 9 - Rede bipartida para sequenciamento preemptivo de tarefas em máquinas paralelas



Fonte: A autora (2020)

Matematicamente pode-se considerar o valor do fluxo em cada arco (i, j) representado por P_{ij} . Uma variável de decisão x_{ij} que representa o fluxo de i para j e uma variável x_{ji} para representar o fluxo de j para i . E representar as capacidades de cada arco (i, j) por c_{ij} .

$$\text{Maximize } \sum_{\{i=s; j \in J\}} P_{ij} \quad (7)$$

subject to

$$\sum_{\{j: (i,j) \in A\}} x_{ij} - \sum_{\{j: (j,i) \in A\}} x_{ji} = \begin{cases} P_{ij} & \text{se } i = s \\ 0 & \forall i \in N - \{s, t\} \\ -P_{ij} & \text{se } i = t \end{cases} \quad (8)$$

bounds

$$0 \leq x_{ij} \leq c_{ij} \quad \forall (i, j) \in A \quad (9)$$

A função objetivo (7) busca maximizar os tempos de processamento das tarefas, representado pelo fluxo que sai de s para o conjunto de J tarefas. O conjunto de restrições formado por (8) garantem a conservação do fluxo máximo em cada um dos nós presentes na rede. A restrição (9) garante que o fluxo não irá ultrapassar a capacidade dos arcos. Com os dados da rede na Figura 9, a modelagem matemática segue como no exemplo a seguir.

$$\text{Maximize } sj1 + sj2 + sj3 + sj4$$

subject to

$$sj1 - j1T_{1,2} - j1T_{3,4} - j1T_{5,7} = 0$$

$$sj2 - j2T_{3,4} - j2T_{5,7} - j2T_{8,9} - j2T_{10,10} - j2T_{11,11} = 0$$

$$sj3 - j3T_{5,7} - j3T_{8,9} - j3T_{10,10} = 0$$

$$sj4 - j4T_{3,4} - j4T_{5,7} - j4T_{8,9} = 0$$

$$j1T_{1,2} - T_{1,2}t = 0$$

$$j1T_{3,4} + j2T_{3,4} + j4T_{3,4} - T_{3,4}t = 0$$

$$j1T_{5,7} + j2T_{5,7} + j3T_{5,7} + j4T_{5,7} - T_{5,7}t = 0$$

$$j2T_{8,9} + j3T_{8,9} + j4T_{8,9} - T_{8,9}t = 0$$

$$j2T_{10,10} + j3T_{10,10} - T_{10,10}t = 0$$

$$j2T_{11,11} - T_{11,11}t = 0$$

bounds

$$0 \leq sj1 \leq 4$$

$$0 \leq sj2 \leq 2,5$$

$$0 \leq sj3 \leq 3,8$$

$$0 \leq sj4 \leq 5$$

$$0 \leq j1T_{1,2} \leq 2$$

$$0 \leq j1T_{3,4} \leq 2$$

$$0 \leq j1T_{5,7} \leq 3$$

$$0 \leq j2T_{3,4} \leq 2$$

$$0 \leq j2T_{5,7} \leq 3$$

$$0 \leq j2T_{8,9} \leq 2$$

$$0 \leq j2T_{10,10} \leq 1$$

$$0 \leq j2T_{11,11} \leq 1$$

$$0 \leq j3T_{5,7} \leq 3$$

$$0 \leq j3T_{8,9} \leq 2$$

$$0 \leq j3T_{10,10} \leq 1$$

$$0 \leq j4T_{3,4} \leq 2$$

$$0 \leq j4T_{5,7} \leq 3$$

$$0 \leq j4T_{8,9} \leq 2$$

$$0 \leq T_{1,2}t \leq 4$$

$$0 \leq T_{3,4}t \leq 4$$

$$0 \leq T_{5,7}t \leq 6$$

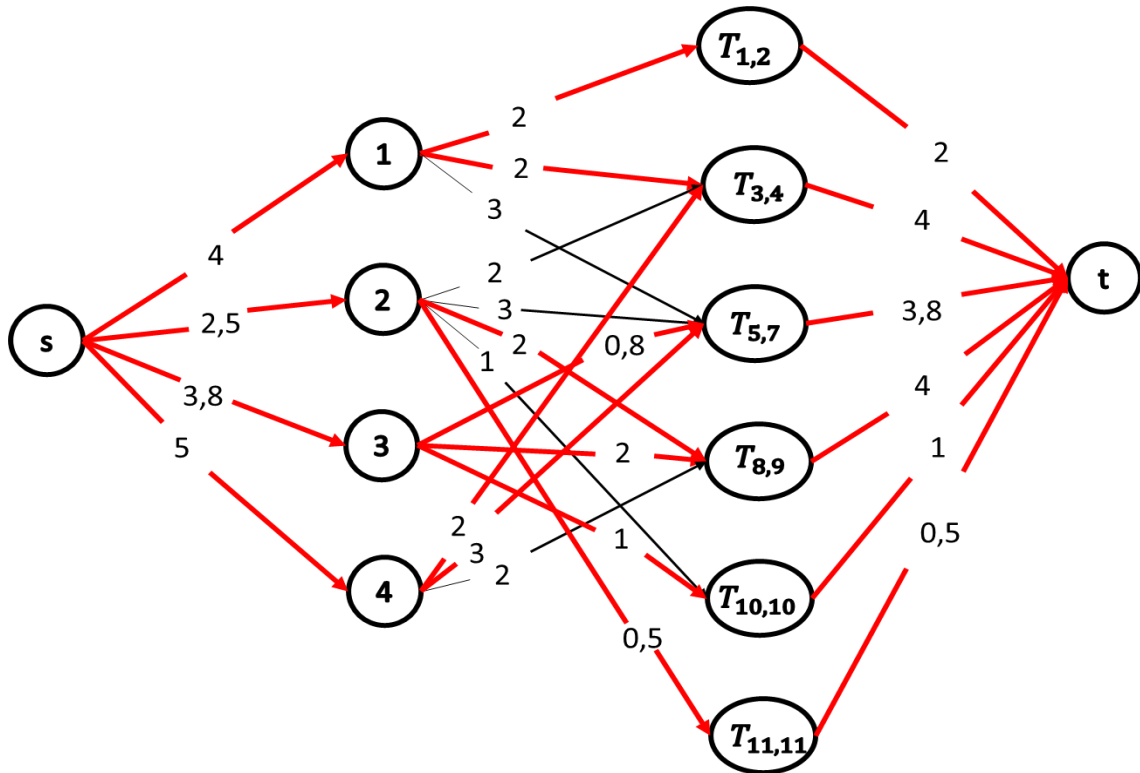
$$0 \leq T_{8,9}t \leq 4$$

$$0 \leq T_{10,10}t \leq 2$$

$$0 \leq T_{11,11}t \leq 2$$

Esta modelagem é escrita em arquivo de extensão “.lp” para ser executado no *SoPlex*. O resultado é exposto na rede resultante que facilita a identificação do sequenciamento encontrado. A rede resultante deste exemplo é vista na Figura 10, os arcos em vermelho representam os arcos em que está passando fluxo.

Figura 10 - Rede resultante para o sequenciamento



Fonte: A autora (2020)

Seguindo a resposta exposta na rede resultante (Figura 10) e partindo do princípio de que os intervalos representam o sequenciamento. A sequência das tarefas encontrada seria: 1, 1, 4, 3, 4, 2, 3, 3 e 2. Este sequenciamento é dito sem atrasos devido o valor do fluxo máximo equivalente ao $\sum P_j = 15,30$ representado, na Figura 10, pela saturação de todo fluxo que sai da origem s .

A maximização dos tempos de processamento obriga as tarefas a serem totalmente concluídas dentro de dois requisitos de sequenciamento: *i*) conclusão antes da data prevista quando é feita a distribuição do processamento de cada tarefa j em pequenos intervalos de tempo $T_{k,l}$, onde o limite superior do intervalo é menor que d_j ; *ii*) a disponibilidade de recursos é respeitada quando é atribuído a quantidade de máquinas à capacidade dos intervalos de tempo $T_{k,l}$.

Em casos onde há um grande número de tarefas (variáveis) há demora em formular e escrever o problema na linguagem de Programação Linear. Como alternativa a este impasse a representação matemática é escrita com auxílio de um programa desenvolvido em *Python* fazendo interface com *SoPlex*.

4.6 PYTHON EM INTERFACE COM SOPLEX

SoPlex é um pacote de otimização, que fornece solução ótima para problemas de Programação Linear (PL) com dados de entrada- racionais. Este pacote é encontrado no *software SCIP (Solving Constraint Integer Programs) Optimization Suite*, mas pode ser usado como um solucionador autônomo de leitura de arquivos no formato “.mps” ou “.lp”, utilizando apenas uma linha de comando (MAHER et al., 2016; LUIS PÉREZ et al., 2016). O *SCIP* trata de resolução de problemas inteiros mistos e programação não-linear restrita (VIGERSKE; GLEIXNER. 2018).

O *Python* é uma linguagem de programação que permite trabalhar mais rapidamente e integrar seus sistemas com mais eficiência. O *Python* suporta uma sintaxe de modelagem intuitiva usando expressões lineares e quadráticas. A interface de *Python* e *SoPlex*, permite a rápida criação de protótipos de novos conceitos algorítmicos (MAHER et al., 2016), por este motivo, tornou-se de interesse para esta pesquisa.

O programa desenvolvido em *Python* converte os dados de entrada em uma rede bipartida $G = (N, A)$ e escreve o modelo matemático apresentado na seção 4.5 que é resolvido no *SoPlex* e convertido em tarefas sequenciadas. Como o sequenciamento não especifica as máquinas correspondente, estas tarefas passam por um processo de designação utilizando o algoritmo de designação de tarefas apresentado na Figura 11, que mantém a sequência de tarefas encontrada no *SoPlex*.

Figura 11 - Algoritmo de designação de tarefas às máquinas

Algoritmo para designação de tarefas às máquinas	
1)	Entrada:
2)	$S_j = S_1, \dots, S_n$ // Conjunto de tarefas sequenciadas
3)	$P_j = (P_1, \dots, P_n), \forall j \in S_j$ // Conjunto dos tempos de processamento de cada tarefa sequenciada
4)	$M_i = M_1, \dots, M_m$ // Conjunto de máquinas disponíveis
5)	$P_i = \emptyset, \forall i \in M$ // Processamento de máquina inicial
6)	Início:
7)	Fase 1 do algoritmo
8)	Para todo $i \in M$:
9)	$M_i \leftarrow S_j[j]$ // Cada máquina i recebe uma tarefa j
10)	$P_i \leftarrow P_j[j]$ // O processamento de máquina i é atualizado
11)	$S_j = S_j - S_j[j]$ // A tarefa alocada j à máquina i é eliminada do conjunto
12)	$P_j = P_j - P_j[j]$ // O tempo de processamento da tarefa alocada j à máquina i é eliminada do conjunto
13)	Para do para
14)	Fim da fase 1
15)	Fase 2 do algoritmo
16)	Enquanto $S_j \neq \emptyset$, faça :
17)	$MenorProcMaq \leftarrow \min(P_i)$ // Para identificar o menor tempo de processamento na máquina i
18)	$M_i[MenorProcMaq] \leftarrow S_j[0]$ // Para alocar a tarefa j à máquina i com o menor processamento
19)	$P_i \leftarrow P_i + P_j[0]$ // O processamento de máquina i é atualizado
20)	$S_j = S_j - S_j[0]$ // Eliminação da tarefa j alocada
21)	$P_j = P_j - P_j[0]$ // Eliminação do tempo de processamento da tarefa j alocada à máquina i
22)	Fim do enquanto
23)	Fim da fase 2
24)	Fim
25)	Saída:
26)	n Tarefas designadas às máquinas

Fonte: A autora (2020)

O algoritmo apresentado na Figura 11 inicia com os dados de entrada presentes entre as linhas 2 e 5. A linha 2 tem o conjunto S_j , que representa as tarefas sequenciadas segundo o resultado encontrado com o *software SoPlex*. A linha 3 contém o conjunto P_j , que representa os tempos de processamento de cada tarefa $j \in S_j$. A linha 4 contém o conjunto de máquinas M_i disponíveis para processamento. A linha 5 contém o conjunto dos tempos de processamento P_i para cada máquina i , que inicialmente é vazio devido à ausência de tarefas sendo processadas nas máquinas.

O algoritmo é dividido em duas fases: na primeira fase é feita a alocação de tarefas às máquinas para que inicie o processamento de máquina. Cada máquina começa com pelo menos uma tarefa; enquanto que na segunda fase é feita a alocação das demais tarefas às máquinas de mais próximas de finalizar o processamento, certificando-se que o sequenciamento está sendo seguido.

A fase 1 inicia com a linha 8 onde diz que para cada máquina, as linhas de comando seguintes devem ser executadas. Linha 9: cada máquina recebe uma tarefa; linha 10: o processamento de

máquina recebe o processamento da tarefa que é designada a máquina. Assim, as máquinas deixam de ter tempo de processamento zero e assumem o tempo de processamento da tarefa atribuída. Linhas 11 e 12 eliminam a tarefa e o processamento, do conjunto de tarefas S_j e processamento P_j que tenha sido alocado à máquina. A fase 1 do algoritmo é, então, finalizada.

A fase 2 inicia com a linha 16 que diz que enquanto houver elementos presentes no conjunto de tarefas sequenciadas S_j , as linhas de comando seguintes devem ser executadas. Na linha 17 é verificado qual o menor tempo de processamento de máquina P_i , a linha 18 verifica qual máquina M_i que tem este menor tempo de processamento P_i e atribui uma tarefa j a esta. Assim, a máquina que tiver o menor tempo de processamento vai receber a próxima tarefa do conjunto de tarefas sequenciadas, que no caso é a primeira tarefa, uma vez que na linha 11 e na linha 20 é verificado que a tarefa é eliminada do conjunto de tarefas ao ser alocado.

Na linha 19, o tempo de processamento da máquina é atualizado somando com o tempo de processamento da tarefa alocada. Assim como na linha 20, a linha 21 repete o mesmo procedimento para o tempo de processamento. A fase 2 do algoritmo é então finalizada. E o algoritmo termina com a linha 25, com o resultado de todas as n tarefas designadas às máquinas. Utilizando o exemplo da rede resultante apresentada na Figura 10 e o algoritmo de designação de tarefas da Figura 11, é possível verificar um exemplo nas 2 máquinas, como é apresentado na Tabela 3.

Tabela 3 – Exemplo de designação de tarefas às máquinas

Saída	Sequenciamento da empresa
Sequência de tarefas J	1, 1, 4, 3, 4, 2, 3, 3, 2.
Designação das tarefas às máquinas	M1: 1, 4, 2, 3, 2 M2: 1, 3, 4, 3

Fonte: A autora (2020)

Depois de feita a designação das tarefas segue-se para apresentação dos possíveis atrasos calculados segundo o *Lateness* (L_j) (equação 2.1) de acordo com a conclusão de cada tarefa (C_j), (Tabela 4).

Tabela 4 – Exemplo resultante do sequenciamento de tarefas

Tarefas j	d_j	Conclusao c_j	$Lateness$
1	8	2,00	-6,00
2	12	7,50	-4,50
3	11	7,80	-3,20
4	10	9,80	-0,20
$L_{máx}$			-0,20

Fonte: A autora (2020)

Como pode ser verificado na Tabela 3, o método utilizado no exemplo não apresentou atrasos. Vale ressaltar que o cálculo *Lateness*, nesta pesquisa, tem a função de somente detectar o atraso do sequenciamento que a empresa utiliza comparado ao que está sendo proposto à mesma.

5 APLICAÇÃO PRÁTICA

Neste capítulo é apresentado uma aplicação prática realizado a partir dos assuntos estudados nesta pesquisa. O principal objetivo deste capítulo é ilustrar a aplicação e os resultados obtidos na utilização do modelo descrito, para um problema de sequenciamento de tarefas em máquinas paralelas uniformes, de uma fábrica de EVA emborrachado, descrevendo os resultados segundo o procedimento utilizado no delineamento deste estudo (seção 4.2).

Ainda neste capítulo é analisado o comportamento do sistema produtivo da empresa e do programa em relação à variação do número de máquinas. O cálculo destas instâncias permite análises de sensibilidade, que demonstram se vale a pena aumentar o número de máquinas ou reduzi-los.

5.1 COLETA DE DADOS

JA empresa que disponibilizou os dados de estudo nesta pesquisa, atua com a fabricação de placas expandidas de EVA (Espuma Vinílica Acetinada). Este tipo de material como *input* no sistema produtivo gera como *outputs*, placas expandidas que são distribuídas nos demais setores para obtenção de produtos finais, tais como: tatames, acessórios para ginástica, colchonetes, tapetes ou até mesmo sola de calçados emborrachados.

As atividades envolvidas no setor de PPCP (Planejamento Programação e Controle da Produção) da empresa incluem programação da produção e a distribuição das operações. A atividade de programação da produção relaciona-se com sequenciamento de tarefas e equilíbrios de processamento nas máquinas. As operações esperam por algum processamento de máquina em uma determinada etapa do fluxo produtivo.

O estudo se concentra apenas no fluxo produtivo das placas expandidas, resumido aqui em seis etapas: *i*) pesagem, *ii*) mistura, *iii*) refinamento, *iv*) corte, *v*) prensagem e *vi*) acabamento. Na primeira etapa, são pesadas as cargas de acordo com as quantidades exigidas nas Ordens de Produção (OPs). Estas cargas são levadas até as máquinas da segunda etapa. As etapas seguintes refinam, cortam e prensam a mistura que sai das máquinas (etapa *ii*), fazendo o acabamento necessário para o produto final. A etapa 2 é a de interesse nesta pesquisa, pois é a etapa em que se

formam as maiores filas de espera com as cargas já pesadas. A regra de priorização *FIFO* (*First In First Out*) é utilizada pela empresa em todas as etapas do processo produtivo.

O ambiente de máquinas deste processo produtivo se assemelha ao de máquinas paralelas uniformes, por terem as mesmas funcionalidades, e diferentes velocidades de processamento das máquinas. Dentre as características de processamento de EVA para a designação de tarefas, cita-se que: cada máquina processa no máximo uma tarefa por vez, as tarefas podem ser processadas em uma máquina por vez e para o sequenciamento o processo permite preempção. Ou seja, o sistema produtivo assume as quatro hipóteses citadas na seção 4.4.

Os dias úteis considerados foram de 30 dias e a jornada de trabalho da empresa é de 24 horas, trabalhando com três turnos devido aos atrasos já existentes no processo produtivo. Os dados liberados pela empresa envolvem as tarefas referentes ao período de junho de 2018 a janeiro de 2019, consideradas pela mesma como tarefas emergenciais, por serem mais críticas em referência a cancelamentos de pedidos por excessivos atrasos na entrega e estarem em estado de reprogramação.

5.2 IMPLEMENTAÇÃO DO MODELO

Para implementação do modelo, os dados coletados foram organizados, resumidamente, como na Tabela 4, em que é especificada a quantidade de tarefas, os tempos de processamento P_j , as datas de início r_j (data de envio da tarefa programada, ao setor de pesagem), as datas devidas d_j (data prometida para a etapa de refinamento, um cliente interno), o número de máquinas e as taxas de velocidades da máquina correspondente.

Os dados referentes ao detalhamento das tarefas (tempo de processamento P_j , datas de início r_j e as datas devidas d_j) são tratados no programa para retornarem: i) o somatório total do tempo de processamento, com a finalidade de identificar a existência de atrasos com o sequenciamento proposto a partir do valor do fluxo máximo e ii) transformar as datas em dias para tratar r_j e d_j como valores inteiros e positivos devido a montagem dos intervalos.

Tabela 5- Dados de entrada para implementação do modelo

Fatores	Dados	Conversão de dados
Tarefas J	46	46
Processamento P_j (dias)	[1; 28]	215
Início das tarefas r_j	[05/06; 09/11]	[1; 31]
Fim das tarefas d_j	[07/08; 18/01]	[16; 309]
Número de máquinas m	9	9
Taxa de velocidade de máquina v_i	[0,2; 0,4]	[0,2; 0,4]

Fonte: A autora (2020)

O programa gerou 23 intervalos de tempo $T_{k,l}$. Foram identificados os nós e as capacidades dos arcos na rede bipartida G . Assim, transformou-se o problema de sequenciamento em um problema de fluxo máximo. Este problema foi introduzido automaticamente no *SoPlex* para encontrar o valor do fluxo máximo do problema, bem como a quantidade de processamento designada a cada elemento do conjunto de intervalos $T_{k,l}$. A saída dos dados e a resposta do *SoPlex* são discutidos na seção 5.3.

5.3 RESULTADOS DO ESTUDO COM DADOS REAIS

Inicia-se discussão dos resultados obtidos a partir dos parâmetros apresentados na Tabela 4, aos quais foram resolvidos de forma exata, pelo modelo de PL no *SoPlex*, para o sequenciamento de tarefas. A solução ótima no *SoPlex* foi obtida em um tempo de 0,01 segundos, com 62 interações. O valor ótimo do fluxo máximo foi de aproximadamente 215 dias.

As respostas obtidas no *SoPlex* mostram que o sequenciamento proposto é viável, ou seja, não apresentou atrasos. O motivo desta afirmação centra-se na solução ótima do modelo, que foi igual ao $\sum P_j$ (Tabela 4) com 215 dias, logo conclui-se que o sequenciamento proposto, nesta pesquisa, não apresenta atrasos. Segue-se, então, para a próxima etapa de resolução do problema, que é a designação de tarefas às máquinas.

O sequenciamento e designação das tarefas da empresa propostos por esta pesquisa é mostrado nas Tabelas 6 e 7, respectivamente. Em ambas as Tabelas, a designação de tarefas nas máquinas seguiu o algoritmo apresentado na seção 4.6. Apesar de o processo produtivo da empresa

permitir preempção, o mesmo não era feito, devido à falta de um ferramental teórico que controlasse a conclusão das mesmas. Assim, as tarefas eram executadas de acordo com a ordem de chegada ao sistema. Na visão da empresa, as tarefas mais antigas devem ser concluídas completamente para iniciar uma tarefa mais recente (regra *FIFO*).

Tabela 6-Sequenciamento e designação de máquinas da empresa

Saída	Sequenciamento da empresa	$C_{máx}$ (dias)
Sequência de tarefas <i>J</i> (FIFO)	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46	34
Designação das tarefas às máquinas	M1: 1, 15	31
	M2: 2, 30, 40, 42, 43	18
	M3: 3, 16, 17, 34, 35, 36, 44, 45	20
	M4: 4, 10, 18, 31	34
	M5: 5, 12, 13, 20, 21, 22, 23, 46	18
	M6: 6, 11, 19, 32	32
	M7: 7, 14, 26, 27, 28, 29, 33, 37, 38, 39, 41	19
	M8: 8, 24	23
	M9: 9, 25	23

Fonte: A autora (2020)

Tabela 7-Sequenciamento e designação de máquinas proposto

Saída	Proposta de sequenciamento	$C_{máx}$ (dias)
Sequência de tarefas <i>J</i>	7, 22, 1, 7, 22, 1, 13, 14, 37, 38, 39, 40, 13, 14, 35, 13, 14, 42, 44, 14, 26, 6, 14, 14, 45, 16, 21, 27, 3, 28, 29, 30, 3, 28, 29, 30, 33, 3, 4, 5, 8, 10, 23, 8, 10, 11, 12, 23, 8, 23, 2, 8, 9, 15, 17, 18, 19, 23, 24, 25, 31, 32, 1, 34, 36, 41, 43, 46	34
Designação das tarefas às máquinas	M1: 7, 35, 13, 29, 8, 8, 15	21
	M2: 22, 14, 3, 10, 18, 34, 36, 46	33
	M3: 1, 14, 33, 11, 23, 31, 27, 61	28
	M4: 7, 42, 26, 6, 45, 16, 27, 28, 30, 8, 19, 43	18
	M5: 22, 44, 14, 9, 32	21
	M6: 1, 14, 21, 29, 5, 12, 8, 24	34
	M7: 13, 3, 3, 23, 25	21
	M8: 14, 28, 4, 10, 23, 17, 1, 41	21
	M9: 37, 38, 39, 40, 13, 14, 30, 23, 2	20

Fonte: A autora (2020)

Quando o processamento de uma tarefa é interrompido com a liberdade de retomar mais tarde em qualquer máquina, é possível adiantar muitas tarefas ao mesmo tempo ao invés de deixá-las

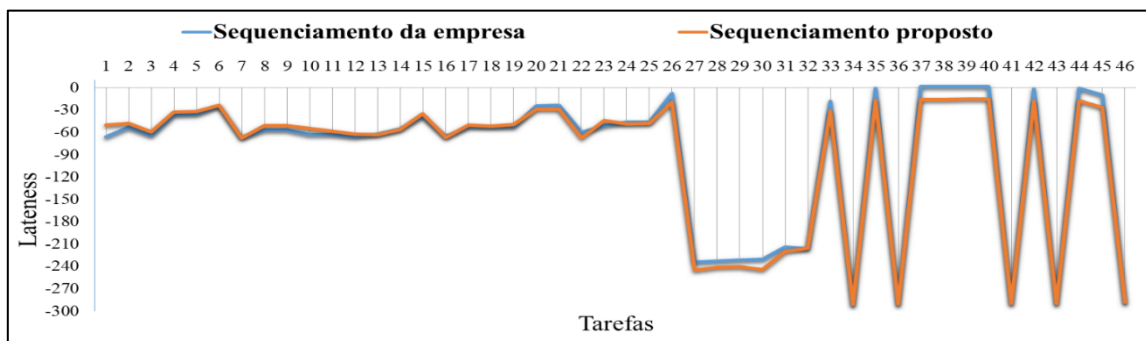
esperando a conclusão de outras tarefas. Vale ressaltar que não foi liberada pela empresa a real programação nem as datas de conclusão das tarefas tendo esta pesquisa que designá-las. E em ambas as designações de tarefas (empresa e proposto) não foram considerados tempos de *setup*.

Sobre o equilíbrio de cargas nas máquinas é possível afirmar que algumas máquinas ficaram ocupadas por mais tempo que outras limitando a programação de novas tarefas, o motivo centra-se no processamento das tarefas, uma vez que algumas apresentam processamentos mais longos que outras tarefas.

Na Tabela 6 é mostrado que apenas duas tarefas (1 e 15) ocupam a máquina 1 por mais de 30 dias. Enquanto as demais máquinas finalizam suas atividades por volta de 20 dias e ficam ociosas comparadas as máquinas 1, 4 e 6 que continuam trabalhando. Ou seja, 66,67% das máquinas são disponibilizadas em um prazo aproximado de 20 dias para novas tarefas enquanto os demais 33,33% permanecem trabalhando por mais 10 dias. No sequenciamento que esta pesquisa propõe (Tabela 7), 77,78% das máquinas finalizam sua execução em torno de 20 dias, podendo estas serem ocupadas com novas tarefas no horizonte de aumento da produtividade.

Uma verificação comparativa entre os sequenciamentos (empresa e proposto) torna-se interessante para averiguar os ganhos significativos que o modelo proporcionou com o sequenciamento e designação das tarefas às máquinas. Esta análise pode ser feita a partir da Figura 12, que mostra graficamente o comportamento do *Lateness* (L_j) entre as tarefas, calculado pela equação 2.1 da seção 2.3.2.

Figura 12-Comportamento do Lateness no sequenciamento da empresa e proposto



Fonte: A autora (2020)

A Figura 12 mostra que o sequenciamento da empresa designado pela regra *FIFO* apresentou, inicialmente, um comportamento bem semelhante ao do sequenciamento proposto, porém em um determinado período o prazo começa a não ser suficiente, como é o caso das tarefas 26, 35, 42, 44 e 45, que tiveram L_j bem próximo de zero. Essa aproximação à data devida acaba induzindo a não conclusão das demais tarefas, resultando em atrasos.

Os atrasos aparecem apenas no sequenciamento da empresa, dentre eles citam-se as tarefas: 37, 38, 39 e 40, que tiveram L_j maior que zero, sendo então responsáveis por atraso equivalente a 9% do total de tarefas programadas para o sequenciamento da empresa. Enquanto que no sequenciamento proposto todas as tarefas foram concluídas com bastante antecipação à entrega.

O detalhamento destes dados é apresentado na Tabela 8, em que a cor azul representa o sequenciamento de tarefas da empresa e a cor laranja representa os dados do sequenciamento proposto. Os dados em tons mais escuros, ilustram o quão longe da data devida, a tarefa concluiu. Os tons mais claros, retratam as tarefas que tiveram seu L_j bem próximo de zero, ou até mesmo, que tiveram atrasos com seu L_j acima de zero. A representação na escala de cores é feita gradativamente.

Tabela 8- Detalhamento de dados do L_j

Tarefas j	d_j	Empresa (FIFO)		Proposto	
		Conclusao c_j	Lateness	Conclusao c_j	Lateness
1	69	2,92	-66,08	18,95	-50,05
2	69	15,83	-53,17	20,86	-48,14
3	68	3,14	-64,86	8,80	-59,20
4	37	0,25	-36,75	3,95	-33,05
5	37	0,83	-36,17	4,59	-32,41
6	25	0,25	-24,75	1,34	-23,66
7	69	1,94	-67,06	1,00	-68,00
8	68	11,11	-56,89	16,90	-51,10
9	68	11,11	-56,89	16,95	-51,05
10	68	5,04	-62,96	12,96	-55,04
11	68	5,04	-62,96	9,38	-58,62
12	68	1,87	-66,13	5,84	-62,16
13	69	6,67	-62,33	6,03	-62,97
14	69	13,06	-55,94	12,30	-56,70
15	69	30,69	-38,31	33,70	-35,30
16	69	3,64	-65,36	1,92	-67,08
17	69	16,50	-52,50	18,86	-50,14
18	69	16,15	-52,85	17,17	-51,83
19	69	16,15	-52,85	19,71	-49,29
20	31	6,86	-24,14	2,11	-28,89
21	31	7,00	-24,00	2,14	-28,86
22	69	9,29	-59,71	1,29	-67,71
23	69	17,92	-51,08	24,64	-44,36
24	69	22,22	-46,78	20,49	-48,51
25	69	22,22	-46,78	20,75	-48,25
26	21	13,07	-7,93	1,09	-19,91
27	248	13,66	-234,34	2,70	-245,30
28	248	15,05	-232,95	6,93	-241,07
29	248	16,44	-231,56	7,36	-240,64
30	248	17,22	-230,78	3,79	-244,21
31	248	33,65	-214,35	27,50	-220,50
32	248	31,99	-216,01	32,73	-215,27
33	36	17,13	-18,87	4,22	-31,78
34	309	17,09	-291,91	17,75	-291,25
35	18	17,11	-0,89	0,97	-17,03
36	309	17,70	-291,30	18,34	-290,66
37	16	17,16	1,16	0,03	-15,97
38	16	17,17	1,17	0,04	-15,96
39	16	17,26	1,26	0,13	-15,87
40	16	17,36	1,36	0,27	-15,73
41	309	18,53	-290,47	19,22	-289,78
42	19	17,44	-1,56	1,08	-17,92
43	309	18,05	-290,95	18,95	-290,05
44	19	17,73	-1,27	1,03	-17,97
45	28	17,82	-10,18	1,42	-26,58
46	309	19,90	-289,10	20,94	-288,06
$L_{\max}$			1,36		-15,73

Fonte: A autora (2020)

Analisando a Figura 12 e a Tabela 8, é possível ver que o $L_{máx}$ no sequenciamento proposto por esta pesquisa, foi de aproximadamente 16 dias de antecipação, enquanto que no da empresa foi de aproximadamente 2 dias, representando um grande ganho, que valida o programa e destaca importância deste estudo. As folgas que a empresa ganhou com tarefas com L_j negativos, no sequenciamento proposto, podem ser utilizadas para aumentar produtividade e a fidelização com o cliente externo, além de melhorar a imagem da empresa.

Para encontrar tais soluções o programa demandou tempo computacional médio de 0,9580 segundos. Neste tempo está incluso: conversão dos dados da empresa em um problema de fluxo máximo, solicitação de resolução do problema no solucionador de programação linear, o SoPlex, conversão da resolução em tarefas sequenciadas e designação de tarefas às máquinas, que especifica cada tarefa em sua máquina correspondente, bem como o tempo de conclusão total ($C_{máx}$) e o *Lateness* ($L_{máx}$).

Como o $L_{máx}$ apresentou uma grande antecipação de finalização das tarefas às datas de entrega (-15,73), torna-se interessante uma análise de sensibilidade no $L_{máx}$ sobre o comportamento do programa, variando o número de máquinas e mantendo os parâmetros atuais da empresa. O comportamento do sistema produtivo e computacional é visto na Tabela 9.

Tabela 9-Análise de sensibilidade (alterando o número de máquinas)

Nº de tarefas	$\sum P_j$	Nº de máquinas	Nº de intervalos T_{kl}	Fluxo Máximo	$L_{máx}$	$C_{máx}$	Tempo de execução do modelo (seg)	Tempo do soplex (seg)
46	215	4	23	104	28,84	97,84	0,906121	0,00
46	215	5	23	215	-14,73	50,72	0,895247	0,01
46	215	6	23	215	-14,85	41,55	0,903696	0,00
46	215	7	23	215	-14,86	37,87	0,904261	0,01
46	215	8	23	215	-14,91	35,00	0,897127	0,01
46	215	9	23	215	-15,73	33,65	0,958008	0,01
46	215	10	23	215	-15,85	31,56	0,920696	0,00
46	215	11	23	215	-15,86	30,10	0,942746	0,01

Fonte: A autora (2020)

Conforme aumentou-se o número de máquinas obteve-se uma leve redução no $L_{máx}$, mantendo o sequenciamento sem atraso a partir de 5 máquinas em operação. De forma análoga, o $C_{máx}$ reduziu significativamente o tempo de conclusão das tarefas, sendo uma redução maior

entre 5 e 9 máquinas. A partir de 9 máquinas a redução do $C_{m\acute{a}x}$ foi bem menor, o que leva a se concluir que aumentar o número de máquinas não traz um ganho que compensa o custo de manter mais máquinas operando, uma vez que diminuir até 5 máquinas mantém o sequenciamento sem atrasos. Do ponto de vista computacional, aumentar o número de máquinas não afeta tanto o tempo do programa o qual foi mantido em menos de 1 segundo. No *SoPlex*, o tempo computacional máximo se manteve em 0,01 segundos.

Em nível operacional, a Tabela 9 mostra que a partir de 5 máquinas operando é possível atender todas as 46 tarefas com um sequenciamento sem atrasos. Trabalhar com apenas 5 máquinas implica em redução de custos com mão de obra, de consumo de energia, ganho de extensão no *layout* da empresa, entre outros benefícios. A designação de tarefas nesta configuração é vista na Tabela 10.

Tabela 10-Análise de sensibilidade (Designação de tarefas em 5 máquinas)

Saída	Proposta de sequenciamento	$C_{m\acute{a}x}$ (dias)
Sequência de tarefas J	7, 22, 1, 7, 22, 1, 13, 14, 37, 38, 39, 40, 13, 14, 35, 13, 14, 42, 44, 14, 26, 6, 14, 14, 45, 16, 19, 20, 21, 19, 27, 3, 19, 28, 29, 30, 3, 19, 28, 29, 30, 33, 3, 4, 5, 19, 10, 19, 23, 10, 11, 12, 19, 23, 23, 2, 8, 9, 15, 17, 18, 23, 24, 25, 31, 32, 1, 34, 36, 41, 43, 46	51
Designação das tarefas às máquinas	M1: 7, 1, 35, 13, 45, 16, 20, 3, 29, 28, 29, 33, 10, 11, 15.	42
	M2: 22, 14, 42, 44, 14, 19, 5, 23, 19, 2, 25, 36, 43, 46	41
	M3: 1, 13, 26, 14, 19, 19, 12, 23, 8, 18, 32	50
	M4: 7, 14, 6, 14, 19, 28, 3, 3, 10, 9, 23, 24, 1, 34, 41	39
	M5: 22, 37, 38, 39, 40, 13, 14, 21, 27, 19, 30, 30, 4, 19, 23, 17, 31	45

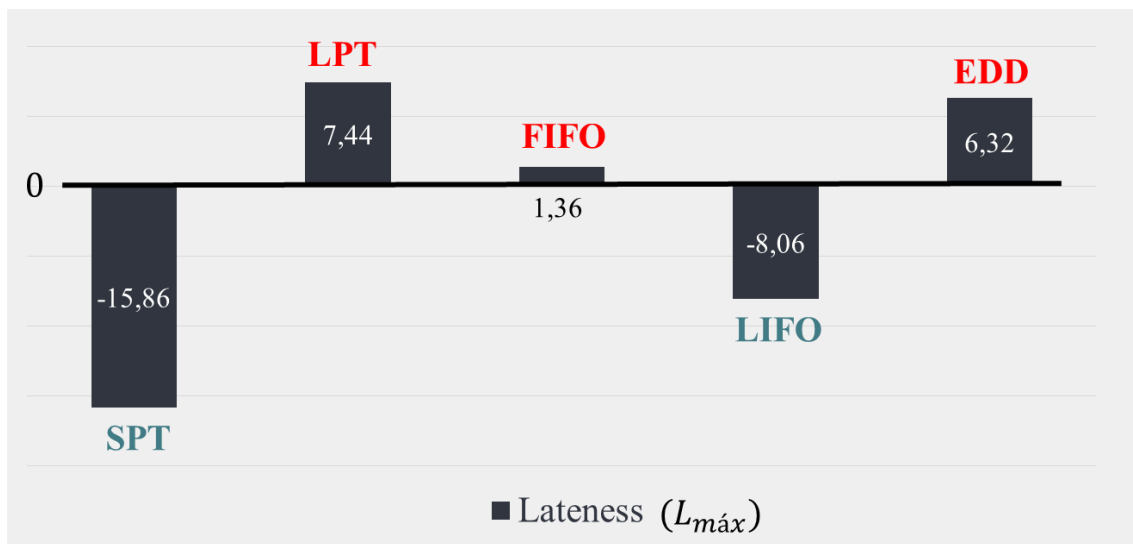
Fonte: A autora (2020)

Como mostra a Tabela 10, operar com 5 máquinas mantém o equilíbrio de cargas nas máquinas, embora tenha aumentado o tempo de conclusão das mesmas. Note que 80% das máquinas concluem as tarefas por volta de 40 dias, enquanto que com 9 máquinas, no sequenciamento proposto, 78% destas concluem em até 20 dias. Ainda assim não há atrasos ($L_j > 0$) em ambos os casos. Os detalhes do cálculo de L_j para esta análise estão no Apêndice 1.

5.4 SEQUENCIAMENTO PROPOSTO X REGRAS DE PRIORIDADES

Nesta subseção são discutidos e analisados os dados calculados com os sequenciamentos originados nas principais regras de prioridades. Vale ressaltar que não foram consideradas preempções e todas as designações foram feitas utilizando o algoritmo desenvolvido por esta pesquisa. A Figura 13 apresenta o $L_{máx}$ correspondente a cada regra, os valores L_j se encontram no Apêndice 2.

Figura 13-Análise com regras de prioridades



Fonte: A autora (2020)

De acordo com a Figura 13 as regras *LPT*, *FIFO* e *EDD* apresentaram atraso, enquanto que as regras *LIFO* e *SPT* apresentaram antecipação de conclusão das tarefas. A regra *SPT* mostrou um $L_{máx}$ menor que o do sequenciamento proposto. Analisando estas regras pelo $L_{máx}$, a *SPT* tem um melhor desempenho comparado ao sequenciamento que esta pesquisa propõe, pela liberdade de aumentar a produtividade. Porém, quanto menor o $L_{máx}$ maior a quantidade de produto aguardando a ser processado na próxima etapa, o que acarreta em custo de estocagem. Neste ponto de vista, a regra *LIFO* se mostra melhor que o sequenciamento proposto e a regra *SPT*, por concluir suas tarefas mais próximas da data devida. Mas outros aspectos devem ser analisados antes de adotar estas regras de priorização. Iniciando por uma análise sobre a regra *SPT*, a Tabela 11 ilustra o sequenciamento e designação.

Tabela 11-Sequenciamento e designação da regra SPT

Saída	Sequenciamento da empresa	$C_{m\acute{a}x}$ (dias)
Sequência de tarefas J (SPT)	26, 38, 35, 37, 44, 42, 45, 39, 40, 21, 20, 4, 6, 16, 34, 27, 36, 43, 33, 5, 12, 41, 28, 29, 30, 7, 46, 22, 1, 3, 10, 11, 13, 23, 8, 9, 14, 18, 19, 24, 25, 17, 2, 32, 31, 15.	42
Designação das tarefas às máquinas	M1: 26, 21, 5, 1, 18	15
	M2: 38, 20, 12, 3, 19	16
	M3: 35, 4, 41, 10, 24	18
	M4: 37, 6, 16, 46, 9, 31	32
	M5: 44, 34, 28, 11, 25	18
	M6: 42, 27, 29, 13, 17	20
	M7: 45, 36, 30, 23, 2	27
	M8: 39, 43, 7, 8, 32	30
	M9: 40, 33, 22, 14, 15	42

Fonte: A autora (2020)

Apesar de a regra *SPT* apresentar um sequenciamento com $L_{m\acute{a}x}$ um pouco melhor que o do sequenciamento proposto, ela tem uma conclusão de 42 dias para entregar todas as tarefas sem atrasos. Enquanto, que o $C_{m\acute{a}x}$ do sequenciamento proposto foi de aproximadamente 34 dias para conclusão, também sem atraso. Ou seja, o sequenciamento proposto permite concluir com uma diferença de 8 dias no $C_{m\acute{a}x}$ com relação a regra *SPT*. A utilização de máquinas em torno de 20 dias é de 77,78% como no sequenciamento proposto, sendo atrativa do ponto de vista de aumento de produtividade, pois grande parte das máquinas são desocupadas em pouco tempo. A próxima regra analisada é a *LPT*, veja a Tabela 12.

Tabela 12- Sequenciamento e designação da regra LPT

Saída	Sequenciamento da empresa	$C_{máx}$ (dias)
Sequência de tarefas J (LPT)	15, 31, 2, 32, 17, 8, 9, 14, 18, 19, 24, 25, 23, 10, 11, 13, 3, 1, 22, 46, 7, 28, 29, 30, 41, 12, 5, 33, 43, 27, 36, 34, 16, 4, 6, 20, 21, 40, 39, 42, 45, 35, 37, 44, 38, 26.	28
Designação das tarefas às máquinas	M1: 15	28
	M2: 31, 3, 28, 41, 40	24
	M3: 2, 11, 7, 27, 20, 45	24
	M4: 32, 13, 29, 30, 44	24
	M5: 17, 10, 1, 46, 36, 21, 35, 26	24
	M6: 8, 19, 5, 4, 39, 37	24
	M7: 9, 24, 33, 16	24
	M8: 14, 25, 43, 34	24
	M9: 18, 23, 22, 12, 6, 42, 38	24

Fonte: A autora (2020)

A regra *LPT* tem um tempo de conclusão bem atrativo para a empresa, mas apresentou sequenciamento inviável, com atrasos de quase 8 dias, apesar de apresentar utilização de máquina de 100% em torno de 20 dias. Esse comportamento é observado pela própria regra, o fato de colocar as tarefas de maior processamento como prioridade afetou, em atrasos, às demais tarefas de menor processamento. Assim, a desvantagem de usar esta regra, neste caso, centra-se no atraso proporcionado pela priorização das tarefas de maior processamento, uma vez que o sequenciamento proposto faz uso de preempção, que permite fracionar os tempos de processamento para não comprometer as datas de entrega negociadas. A próxima regra analisada é a LIFO, veja a Tabela 13.

Tabela 13- Sequenciamento e designação da regra LIFO

Saída	Sequenciamento da empresa	$C_{máx}$ (dias)
Sequência de tarefas J (LIFO)	46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1.	34
Designação das tarefas às máquinas	M1: 46, 19, 11	18
	M2: 45, 33, 26, 25, 13, 6, 5, 3	21
	M3: 44, 36, 28, 20, 17, 8	26
	M4: 43, 29, 21, 18, 10	18
	M5: 42, 32, 7, 1	21
	M6: 41, 23, 14	21
	M7: 40, 30, 22, 16, 15	32
	M8: 39, 31, 4, 2	34
	M9: 38, 37, 35, 34, 27, 24, 12, 9	21

Fonte: A autora (2020)

A regra *LIFO* teve um comportamento bem semelhante ao do sequenciamento proposto inclusive quando comparado o a utilização de máquinas que foi de 77,78% para conclusão das tarefas em torno de 20 dias. Pela Tabela 13 é visto que o tempo de conclusão foi o mesmo, a utilização das máquinas também coincide. Devido estas semelhanças, o sequenciamento da regra *LIFO* foi calculado novamente, mas para o caso de 5 máquinas, a fim de verificar diferenças entre ambos sequenciamentos (*LIFO* e proposto). A Tabela 14 trata do sequenciamento e designação das tarefas com a regra *LIFO* para 5 máquinas.

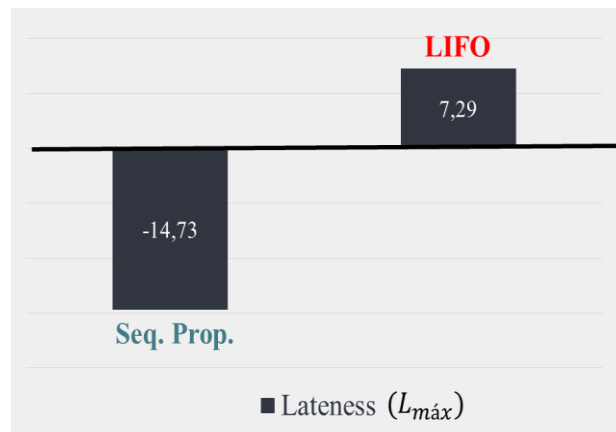
Tabela 14- Sequenciamento e designação da regra LIFO para 5 máquinas

Saída	Sequenciamento da empresa	$C_{máx}$ (dias)
Sequência de tarefas J (LIFO – 5 máquinas)	46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1.	50
Designação das tarefas às máquinas	M1: 46, 28, 23, 22, 18, 12, 11, 7, 31.	40
	M2: 45, 40, 35, 34, 31, 14, 9.	41
	M3: 44, 41, 30, 27, 24, 17, 10, 6, 5, 4, 2.	50
	M4: 43, 33, 29, 26, 25, 21, 20, 19, 13, 8.	41
	M5: 42, 39, 38, 37, 36, 32, 16, 15.	45

Fonte: A autora (2020)

Em ótica ao tempo de conclusão total das tarefas a diferença foi de 1 dia, o sequenciamento proposto chegou a 51 dias e o sequenciamento da *LIFO* foi de 50 dias. Em relação a utilização dos recursos, pode-se afirmar que o resultado foi bem parecido, ambos apresentaram 80% das máquinas ficam livres para novas tarefas em aproximadamente 40 dias. A diferença significativa que houve entre os dois sequenciamentos foi referente ao $L_{máx}$ para o ambiente de 5 máquinas (Figura 14) . O sequenciamento proposto teve um $L_{máx}$ de aproximadamente 15 dias antecipados à data devida (Tabela 8 da subseção 5.4.1) e o sequenciamento *LIFO* apresentou atraso de até 8 dias, aproximadamente, ver Apêndice 3.

Figura 14-Análise entre sequenciamento proposto e LIFO (5 máquinas)



Fonte: A autora (2020)

A regra LIFO foi bem atrativa no ambiente de 9 máquinas, porém quando reduzida a 5 máquinas, as tarefas sofrem atrasos saindo do escopo de interesse. Vale ressaltar que a redução de máquinas não comprometeu tanto $L_{máx}$ quanto com a regra *LIFO*. A última regra analisada é a EDD, veja a Tabela 15.

Tabela 15- Sequenciamento e designação da regra EDD

Saída	Sequenciamento da empresa	$C_{máx}$ (dias)
Sequência de tarefas J (EDD)	40, 39, 38, 37, 35, 44, 42, 26, 6, 45, 21, 20, 33, 5, 4, 12, 11, 10, 9, 8, 3, 25, 24, 23, 22, 19, 18, 17, 16, 15, 14, 13, 7, 2, 1, 32, 31, 30, 29, 28, 27, 46, 43, 41, 36, 34.	34
Designação das tarefas às máquinas	M1: 37, 33, 2, 32	33
	M2: 38, 20, 11, 19, 31	34
	M3: 39, 3, 16, 17, 34, 41	19
	M4: 40, 9, 25	23
	M5: 35, 4, 1, 15	31
	M6: 42, 5, 7, 14, 27, 28, 29, 43	18
	M7: 44, 21, 10, 18, 30, 46	20
	M8: 26, 45, 8, 24	23
	M9: 6, 12, 13, 22, 23, 36	18

Fonte: A autora (2020)

A regra *EDD* teve seu tempo de conclusão semelhante ao do sequenciamento proposto, mas com relação a utilização de máquinas 66,67% destas se mantiveram ocupadas em torno de 20 dias, como no sequenciamento executado pela empresa que utiliza a regra *FIFO*. Por outro lado, sequenciar as 46 tarefas com a regra *EDD*, acarretou em aproximadamente 7 dias de atraso, o que demonstra desvantagem comparada às técnicas que este estudo utilizou.

Em resumo estas análises para o cenário de 9 máquinas foram postas na Tabela 16 fazendo referência aos aspectos destacados, tais como: atraso máximo, conclusão máximo e utilização de máquinas que representa a liberação das máquinas em torno de 20 dias possibilitando o aumento de produtividade com a inclusão de mais tarefas nas máquinas liberadas.

Quadro 3 – Resumo da avaliação comparativa quanto aos aspectos

Regras	Aspectos		
	$L_{máx}$	Tempo total de conclusão ($C_{máx}$)	Utilização de máquinas
Sequenciamento proposto	-15,73	34	77,78%
SPT	-15,86	42	77,78%
LPT	7,44	28	100%
LIFO	-8,06	34	77,78%
EDD	6,32	34	66,67%

Fonte: A autora (2020)

Portanto, o sequenciamento proposto neste estudo continua melhor do que utilizar as regras de prioridades aqui discutidas em todos os aspectos. Vale ressaltar que a regra LIFO apresentou boas soluções comparada ao sequenciamento proposto no cenário de 9 máquinas, mas a análise para o cenário de 5 máquinas não foi tão atrativa pois apresentou atraso de até 8 dias. Os resultados obtidos nesta pesquisa, com aplicação real, mostram-se mais eficaz no que tange as desvantagens destacadas em cada regra de priorização.

6 CONSIDERAÇÕES FINAIS

Em todo sistema produtivo, seja de serviço ou manufatura, o conhecimento das técnicas de sequenciamento de tarefas pode levar à redução de custos com atrasos. No serviço, sequenciam-se operações, enquanto que na manufatura, são sequenciadas OPs. Assim, esta pesquisa enfatiza em como sequenciar da forma mais eficiente possível, utilizando otimização combinatória.

O objetivo desta pesquisa consistiu em propor uma solução ao problema de sequenciamento e de designação de tarefas às máquinas paralelas uniformes, utilizando dados reais e preempção. Para isto uma abordagem exata é utilizada ao tratar o problema de sequenciamento como um problema de fluxo máximo. Para tratar do problema de designação de tarefas um algoritmo foi proposto. Para alcance deste objetivo, o modelo exato utilizado nesta pesquisa, baseou-se em uma aplicação demonstrada por Ahuja, Magnanti e Orlin (1993), que faz uso da modelagem proposta por Federgruen e Groenevelt (1986).

A ordenação das tarefas (sequenciamento) apresentou resultados bem significativos em relação ao cenário atual da empresa estudada. O estudo também demonstrou uma melhor distribuição das tarefas alocadas às máquinas. Do ponto de vista computacional, a solução obtida no programa foi alcançada em um tempo médio de 0,96 segundos, o que é bastante atrativo na prática, pois problemas de sequenciamento são operacionais e precisam ser resolvidos frequentemente.

O sequenciamento proposto não apresentou atrasos, as tarefas foram concluídas com antecipação de aproximadamente 16 dias, o que representa uma folga que permite aumentar a produtividade incluindo mais tarefas a serem processadas. Enquanto que o sequenciamento da empresa sofreu atraso máximo de aproximadamente 2 dias representado em 9% das tarefas programadas. Vale ressaltar que nem todas as tarefas da programação da empresa foram liberadas, apenas as emergenciais. Ou seja, se com apenas 46 tarefas o sequenciamento da empresa apresentou atrasos, imagine com todas as tarefas da programação o tamanho do atraso, assim a antecipação pode ser preenchida com as demais tarefas que não participaram do estudo. Sendo interessante a análise com as demais tarefas.

O programa ajudou no detalhamento dos dados comparativos para análise dos ganhos que o sequenciamento e designação das tarefas proporcionou para a empresa e na busca pelos motivos que deixavam o sequenciamento usual da empresa com atrasos. A justificativa para este problema

está vinculada ao uso de preempção. Quando o processamento de uma tarefa é interrompido com a liberdade de retomar mais tarde em qualquer máquina, é possível adiantar muitas tarefas ao mesmo tempo invés de esperar a conclusão de uma para iniciar o processamento de outra.

Foram feitas análises de sensibilidades variando o número de máquinas entre 4 e 12. Conforme aumentou-se o número de máquinas obteve-se uma leve redução no $L_{máx}$ ao tempo que o $C_{máx}$ reduziu significativamente. Concluiu-se que acrescentar ou reduzir a quantidade de máquinas não influenciou muito no tempo computacional do programa, mantendo-se em menos de 1 segundo. Do ponto de vista de ganhos para a empresa, foi concluído que operar o sistema produtivo reduzindo de 9 para 5 máquinas seria suficiente para atender todas as 46 tarefas conservando o sequenciamento sem atrasos. Além de implicar em redução de custos com mão de obra, consumo de energia, ganho de extensão no *layout* da empresa, entre outros benefícios. Por outro lado, a conclusão foi prolongada de 20 dias para 40 dias em aproximadamente 80% das máquinas.

Nesta pesquisa, também foram feitas comparações das regras de prioridades comumente discutidas na literatura e o sequenciamento proposto. Foram verificados os aspectos relacionados a utilização dos recursos, ociosidade de máquina e tempo total de conclusão. Os resultados obtidos com a aplicação prática mostram-se mais eficaz que as desvantagens apontadas no estudo destes dados utilizados nas regras de prioridades.

As principais dificuldades enfrentadas no desenvolvimento deste trabalho, incluem: 1) a obtenção de dados para teste do programa desenvolvido e 2) a procura por trabalhos recentes que tratam da redução de sequenciamento para problemas de fluxo em redes. O primeiro caso é justificado pela baixa quantidade de empresas interessadas em contribuir com o estudo. O segundo caso é devido à escassez de trabalhos atuais relacionados.

Como limitações para esta pesquisa, cita-se que não foi possível a obtenção de uma maior quantidade de dados, pois a empresa liberou apenas as tarefas consideradas emergenciais e também não liberou as datas de conclusões das tarefas, tendo esta pesquisa que designar as tarefas às máquinas. Vale ressaltar que esta pesquisa não considerou nenhum peso relacionado a clientes de maior prioridade.

Para trabalhos futuros, recomenda-se testar esta aplicação em outros sistemas produtivos de manufatura ou serviço, para verificar o comportamento dos dados em diferentes ambientes de produção. Recomenda-se também, incluir outros recursos, como: tempo de *setup*, fazendo uso de

técnicas de fluxo em redes análogas a este trabalho. Uma outra sugestão para trabalhos futuros, seria verificar a distribuição de probabilidade que os tempos de processamentos de um determinado sistema segue, ou seja, considerar os tempos de processamento estocásticos, para analisar os impactos no método proposto.

Outra proposta para trabalhos futuros consiste em tornar aleatória a escolha inicial das máquinas as quais devem ser alocadas as tarefas no algoritmo de designação de tarefas proposto nesta pesquisa, com a finalidade de verificar a influência desta mudança no tempo de conclusão das tarefas e na comparação com as regras de priorização utilizando também o algoritmo, uma vez que as diferenças nas taxas de velocidade das máquinas podem ser consideradas um fator de grande influência.

REFERÊNCIAS

ABREU, J. C.; PEREIRA, A. A. S. Meta-heurística multiobjetivo para sequenciamento de máquinas paralelas não relacionadas com tempos de preparação dependentes da sequência. *Revista Científica FAGOC-Multidisciplinar*, v. 3, n.1, p. 31-43, 2019.

AHUJA, R. K.; MAGNANTI, T. L.; ORLIN, J. B. *Network flows: theory, algorithms, and applications*. Upper Saddle River, New Jersey: Prentice Hall, 1993.

ALBUQUERQUE, C. et al. Análise do planejamento e controle da produção em ambiente sob demanda no setor de confecção industrial: estudo de caso. *REA-Revista Eletrônica de Administração*, v. 16, n. 2, p. 336-351, 2018.

ALCOFORADO, L. F.; BORNSTEIN, C. T. Algoritmo para realocação de tarefas às máquinas para um problema de sequenciamento em máquinas paralelas. In: Simpósio Brasileiro de Pesquisa Operacional – SBPO, Rio de Janeiro/RJ, 2002

ARENALES, M. et al. *Pesquisa Operacional*. Rio de Janeiro: Elsevier: ABEPRO, 2011.

BAPTISTE, P. et al. The complexity of mean flow time scheduling problems with release times. *Journal of Scheduling*, v. 10, n. 2, p. 139-146, 2007.

BARD, J. F.; WAN, L. The task assignment problem for unrestricted movement between workstation groups. *Journal of Scheduling, USA*, v. 9, p. 315-341, 2006

BAZARAA, M. S.; JARVIS, J. J.; SHERALI, H. D. *Linear Programming and Network Flows*. New Jersey: John Wiley and Sons. Inc., Hoboken, 2010.

BENDER, M. A.; RABIN, M. O. Online scheduling of parallel programs on heterogeneous systems with applications to cilk. *Theory of Computing Systems*, v. 35, n. 3, p. 289-304, 2002.

CHERRI, L. H.; MUNDIM, L. R.; CHERRI, A. C.; VIANNA, A. C. G. Designação de veículos para transporte de carga a centros de distribuição. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, v. 7, n. 1, 2020.

COTA, L. P. *Abordagens exatas e heurísticas para o problema de sequenciamento em máquinas paralelas não relacionadas com tempos de preparação dependentes da sequência*. Tese (Doutorado). Universidade Federal de Minas Gerais. 134 f. 2018.

DONADONI, I. G. D.; PEREIRA, A. A. S. Regra de prioridade para o problema de Flow Shop híbrido com máquinas paralelas não relacionadas. *Revista Científica Fagoc Multidisciplinar*, v. 2, n. 2, 2017.

ESTENDER, A. C. et al. A importância do Planejamento e Controle da produção. In: Simpósio Internacional de Gestão de Projetos–SINGEP, 2017.

FEDERGRUEN, A.; GROENEVELT, H. Preemptive scheduling of uniform machines by ordinary network flow techniques. *Management Science*, v. 32, n. 3, p. 341-349, 1986.

FERREIRA, G. D. S. *Algoritmos genéticos adaptativos para solucionar problemas de sequenciamento do tipo job-shop flexível*. Dissertação (Mestrado). Programa de Pós-graduação em modelagem computacional. Universidade Federal de Juiz de Fora. Juiz de Fora-MG, 2018.

FOHLER, G. How different are offline and online scheduling? In: II International Real Time Scheduling Open Problems Seminar, Beto, Portugal, p. 5-6, 2011.

FUCHIGAMI, H. Y.; RANGEL, S. Uma análise de estudos de casos em sequenciamento da produção. In: XLVI Simpósio Brasileiro em Pesquisa Operacional, Salvador-BA. SOBRAPO, 2014.

GANGA, G. M. D. *Trabalho de conclusão de curso (TCC) na engenharia de produção: um guia prático de conteúdo e forma*. São Paulo: Atlas, 2012.

GERHARDT, T. E.; SILVEIRA, D. T. *Métodos de pesquisa*. Plageder, 2009.

GIL, A. C. *Como elaborar projetos de pesquisa*. 5.ed. São Paulo: Atlas, 2010.

GONÇALVES, D. V. et al. Programação de materiais e controle da produção. *Revista da Mostra de Iniciação Científica (RMIC)*, v. 1, n. 1, 2018.

GONZALEZ, T.; SAHNI, S. Preemptive scheduling of uniform processor systems. *Journal of the ACM (JACM)*, v. 25, n. 1, p. 92-101, 1978.

GRÖTSCHER, M.; LOVÁSZ, L.; SCHRIJVER, A. *Algoritmos geométricos e otimização combinatória*. Springer Science & Business Media, ed. 2, 2012.

GUERRINI, F. M.; BELHOT, R. V.; AZZOLINNI JÚNIOR, W. *Planejamento e controle da produção: Modelagem e implementação*. Elsevier, 2 ed. 328p. 2018.

HABIBA, H. et al. Minimizing Makespan on Identical Parallel Machines. *Electrotehnica, Electronica, Automatica*, v. 66, n. 1, p. 180-185, 2018.

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

HITOMI, K. *Manufacturing Systems Engineering: a unified approach to manufacturing technology, production management and industrial economics*. 2 ed., Londres: Taylor & Francis, 2017.

HORN, W. Some simple scheduling algorithms. *Naval Research Logistics Quarterly*. v.21, n. 1, p. 177-185, 1974.

HOSSEINABADI, A. A. R. et al. Extended genetic algorithm for solving open-shop scheduling problem. *Soft computing*, v. 23, n. 13, p. 5099-5116, 2019.

KAHRAMAN, C.; YAVUZ, M. (Ed.). *Production Engineering and Management under fuzziness*. Verlag Berlin Heidelberg: Springer, 2010.

KAYVANFAR, V.; ZANDIEH, M.; TEYMOURIAN, E. Um algoritmo inteligente de gota d'água para programação de máquinas paralelas idênticas com tempos de processamento controláveis: uma abordagem Just-In-Time. *Matemática Computacional e Aplicada*, v. 36, n. 1, p. 159-184, 2017.

KELLER, M. T.; TROTTER, W. T. *Applied Combinatorics*. Georgia Institute of Technology: Preliminary Edition, 2015.

KIM, E. Scheduling of uniform parallel machines with s-precedence constraints. *Mathematical and Computer Modelling*, v. 54, p. 576–583, 2011.

KOULAMAS, C.; KYPARISIS, G. J. Um algoritmo LPT modificado para o problema uniforme de minimização de produção de duas máquinas paralelas. *European Journal of Operational Research*, v. 196, n. 1, p. 61-68, 2009.

LEE, K.; LEUNG, J. Y. T.; PINEDO, M. L. Makespan minimization in online scheduling with machine eligibility. *4OR*, v. 8, n. 4, p. 331-364, 2010.

LI, K.; XIAO, W.; YANG, S. Minimizing total tardiness on two uniform parallel machines considering a cost constraint. *Expert Systems with Applications*, v. 123, p. 143-153, 2019.

- LI, K. et al. Uniform parallel machine scheduling with fuzzy processing times under resource consumption constraint. *Applied Soft Computing*, v. 82, p. 105585, 2019.
- LUIS, P. A. Linear Programming with SoPlex, SoPlex complexity and SCIP complexity. 2016.
- LUSHCHAKOVA, I. N. Preemptive scheduling of equal length jobs with release dates on two uniform parallel machines. *Discrete Optimization*, v. 6, p.446-460, 2009.
- LUSTOSA, L. J et al. *Planejamento e controle da produção*. Elsevier Brasil, 2008.
- MAHER, S. et al. PySCIPOpt: Mathematical programming in python with the SCIP optimization suite. In: International Congress on Mathematical Software. *Springer, Cham*. p. 301-307, 2016.
- MARTEL, C. Preemptive scheduling with release times, deadlines, and due times. *Journal of ACM*, v. 29, p. 812-829, 1982.
- MARTÍ, R. REINELT, G. The Linear Ordering Problem - Exact and Heuristic Methods in Combinatorial Optimization, Heidelberg Dordrecht London New York: *Springer*, vol. 175, 2011.
- MCDONALD, T.; ELLIS, K. P.; VAN AKEN, E. M.; KOELLING, C. P. Development and application of a worker assignment model to evaluate a lean manufacturing cell. *International Journal of Production Research*, UK, v. 47, n.9, p. 2427-2447, 2009.
- MENDIS, H. R.; AUDSLEY, N. C.; SOARES INDRUSIAK, L. Dynamic and static task allocation for hard real-time video stream decoding on NoCs. LITES: *Leibniz Transactions on Embedded Systems*, v. 4, n. 2, p. 1-25, 2017.
- MENG, T.; PAN, Q.; LI, J.; SANG, H. An improved migrating birds optimization for an integrated lot-streaming Flow Shop scheduling problem. *Swarm and Evolutionary Computation*, v. 38, p. 64-78, 2018.
- MOREIRA, D. *Administração da Produção e Operações*. São Paulo: Cengage Learning, 2011.
- MUNDIM, L. R.; FUCHIGAMI, H. Y. Uma heurística robusta para programação de máquinas paralelas com tempos de setup dependentes da sequência. *Revista Produção Online*, v. 17, n. 2, p. 463-481, 2017.
- NETTO, P. O. B.; JURKIEWICZ, S. *Grafos: introdução e prática*. Editora Blucher, ed. 2, 2017.

NIU, S. et al. Distributionally robust single machine scheduling with the total tardiness criterion. *Computers & Operations Research*, v. 101, p. 13-28, 2019.

NOUIRI, M. et al. An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. *Journal of Intelligent Manufacturing*, v. 29, n. 3, p. 603-615, 2018.

OLIVEIRA, G. B. *Planejamento e Controle da Produção*: Um estudo de caso em uma indústria de alimentos no município de Cacoal – Rondônia. Monografia em Engenharia de Produção da Universidade Federal de Rondônia, Cacoal-RO, 57f., 2017.

ÖNER-KÖZEN, M.; MINNER, S. Impact of priority sequencing decisions on on-time probability and expected tardiness of orders in make-to-order production systems with external due-dates. *European Journal of Operational Research*, v. 263, n. 2, p. 524-539, 2017.

OTTEN, M.; BRAAKSMA, A.; BOUCHERIE, R. J. Minimizing Earliness/Tardiness costs on multiple machines with an application to surgery scheduling. *Operations research for health care*, v. 22, 2019.

PARIS, Q. *An economic interpretation of linear programming*. Springer, 2016.

PINEDO, M. L. *Scheduling: theory, algorithms and systems*. New York: Springer, 5 ed., 2016

PINEDO, M. L. *Scheduling*. New York: Springer, 4. ed., 2012.

PRESTES, E. *Introdução à Teoria dos Grafos*. Universidade Federal do Rio Grande do Sul, Instituto de Informática, Departamento de Informática Teórica, Tech. Rep, 2016.

SAHNI, S.; CHO, Y. Nearly online scheduling of a uniform processor system with release times. *SIAM Journal on Computing*, v.8, n.2, p. 275-285, 1979.

SANTOS, T. S.; SANTOS, F. S. A influência das rotas no planejamento, programação e controle da produção de uma cerâmica. *Brazilian Journal of Production Engineering-BJPE*, v. 4, n. 3, p. 94-112, 2018.

SCHÄFFERSA, P.; MÜTZEA, A.; NYHUISA, P. Integrated Concept for Acquisition and Utilization of Production Feedback Data to Support Production Planning and Control in the Age of Digitalization. *Procedia Manufacturing*, v. 31, p. 225-231, 2019.

SCHRIJVER, A. *A course in combinatorial optimization*. Amsterdam: Springer, 2017.

SCHRIJVER, A. *Otimização combinatória: poliedros e eficiência*. Springer Science & Business Media, v. 24, 2003.

SHAKHLEVICH, N. V.; STRUSEVICH, V. A. Preemptive scheduling on uniform parallel machines with controllable job processing times. *Algorithmic*, v. 51, n. 4, p. 451-473, 2008.

SHIOURA, A.; SHAKHLEVICH, N. V.; STRUSEVICH, V. A. Preemptive models of scheduling with controllable processing times and of scheduling with imprecise computation: A review of solution approaches. *European Journal of Operational Research*, v. 266, n. 3, p. 795-818, 2018.

SILVA, C. A.; PACHECO, D. A.; CUNHA JÚNIOR, J. J. Problema integrado de sequenciamento da produção e dimensionamento de lotes em um processo de injeção de termoplásticos. In: L Simpósio Brasileiro em Pesquisa Operacional, Rio de Janeiro-RJ. SOBRAPO, 2018.

SILVA, T. C.; FUCHIGAMI, H. Y. Desenvolvimento e análise de regras de sequenciamento da produção para minimização de medidas de atraso em Flow Shop com tempos de setup. *Revista Produção Online*. Florianópolis, SC, v. 18, n. 2, p. 454-478, 2018.

SILVA, W. R.; ESTENDER, A. C.; BARBOSA, L. Implantação do sistema de PCP em micro, pequenas e médias empresas. *REGIT*, v. 4, n. 2, 2015.

SITTERS, R.; Yang, L. A $(2 + \epsilon)$ - approximation for precedence constrained single machine scheduling with release dates and total weighted completion time objective. *Operations Research Letters*, v. 46, n. 4, p. 438-442, 2018.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. *Administração da Produção*. São Paulo: Atlas, 3 ed, 2009.

SOPER, A. J.; STRUSEVICH, V. A. Schedules with a single preemption on uniform parallel machines. *Discrete Applied Mathematics*, v. 261, p. 332-343, 2019.

TAVARES, H. M. G. *Estudo e Análise do Sequenciamento de Tarefas de Produção: Job Shop scheduling*. Dissertação (Mestrado). Programa de Pós-Graduação em Engenharia e Gestão Industrial, Instituto Politécnico do Porto, Vila do Conde, Portugal, 2015.

TUBINO, D. F. *Planejamento e Controle da Produção: Teoria e Prática*. São Paulo: Atlas, 2007.

TUCKER, A. *Applied combinatorics*. New York: Wiley, 6 ed., 2012.

VERÍSSIMO, J. M. F. *Loteamento e sequenciamento da produção: um caso de estudo na indústria de injeção de plásticos*. Dissertação (Mestrado), Programa de Pós-graduação em Engenharia e Gestão Industrial, Universidade de Coimbra, 2016.

VIEGAS, V. A. *O problema de sequenciamento de tarefas em máquinas paralelas: um estudo de caso em um departamento de credenciamento*. Monografia em Engenharia de Produção, Universidade Federal Fluminense, Niterói/RJ, 2016.

VIGERSKE, S.; GLEIXNER, A. SCIP: Global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. *Optimization Methods and Software*, v. 33, n. 3, p. 563-593, 2018.

WOEGINGER, G. J. The Open Shop Scheduling Problem. In: 35th Symposium on Theoretical Aspects of Computer Science (STACS 2018). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

WOLSEY, L. A.; NEMHAUSER, G. L. *Integer and combinatorial optimization*. John Wiley & Sons, 2014.

XAVIER, A. S.; LINS, S. L. S.; SILVA, A. A. *Sequenciamento de tarefas em máquinas paralelas uniformes em processo de fabricação de EVA e Calçados*. In: LI Simpósio Brasileiro de Pesquisa Operacional, Limeira-SP. SOBRAPO, 2019.

XIAO, L.; SONG, S.; CHEN, X.; COIT, D. W. Joint optimization of production scheduling and machine group preventive maintenance. *Reliability Engineering & System Safety*, v. 146, p. 68-78, 2016.

YALAOUI, N. et al. Fuzzy Project Scheduling. In: Kahraman C., Yavuz M. (Eds) *Engenharia e Gerenciamento de Produção sob Fuzziness. Estudos em Fuzziness e Soft Computing*, Springer, Berlin, Heidelberg, v. 252, p. 143-170, 2010).

ZAREI, M.; DAVVARI, M.; KOLAHAN, F.; WONG, K. Simultaneous selection and scheduling with sequence-dependent setup times, lateness penalties, and machine availability constraint: Heuristic approaches. *International Journal of Industrial Engineering Computations*, v. 7, n. 1, p. 147-160, 2016.

ZHOU, S.; LIU, M.; CHEN, H.; LI, X. An effective discrete differential evolution algorithm for scheduling uniform parallel batch processing machines with non-identical capacities and arbitrary job sizes. *International Journal of Production Economics*, v. 179, p. 1-11, 2016.

ZOLFAGHARI, S.; EL-BOURI, A.; NAMIRANIAN, B. Heuristics for Large Scale Labour Scheduling Problems in Retail Sector. INFOR, Canadá, v. 45 p. 111-122, 2007.

APÊNDICE A – AMBIENTE DE 5 MÁQUINAS ($L_{m\acute{a}x}$)

j	d_j	C_j	L_j
1	69	40,00	-29,00
2	69	26,17	-42,83
3	68	15,86	-52,14
4	37	7,64	-29,36
5	37	8,23	-28,77
6	25	3,25	-21,75
7	69	1,00	-68,00
8	68	22,16	-45,84
9	68	23,40	-44,60
10	68	21,21	-46,79
11	68	13,71	-54,29
12	68	10,05	-57,95
13	69	8,95	-60,05
14	69	13,76	-55,24
15	69	41,49	-27,51
16	69	4,26	-64,74
17	69	27,50	-41,50
18	69	33,27	-35,73
19	69	43,00	-26,00
20	31	4,45	-26,55
21	31	4,41	-26,59
22	69	1,29	-67,71
23	69	50,72	-18,28
24	69	36,14	-32,86
25	69	37,28	-31,72
26	21	3,01	-17,99
27	248	27,00	-221,00
28	248	13,20	-234,80
29	248	13,68	-234,32
30	248	14,39	-233,61
31	248	45,00	-203,00
32	248	49,10	-198,90
33	36	7,92	-28,08
34	309	37,72	-271,28
35	18	1,89	-16,11
36	309	37,87	-271,13
37	16	1,03	-14,97
38	16	1,04	-14,96
39	16	1,13	-14,87
40	16	1,27	-14,73
41	309	38,99	-270,01
42	19	2,37	-16,63
43	309	38,48	-270,52
44	19	2,40	-16,60
45	28	3,76	-24,24
46	309	40,56	-268,44
$L_{m\acute{a}x}$			-14,73

APÊNDICE B –PRINCIPAIS REGRAS DE PRIORIDADES ($L_{m\acute{A}x}$)

j	d_j	SPT		LPT		LIFO		EDD	
		C_j	L_j	C_j	L_j	C_j	L_j	C_j	L_j
1	69	3,90	-65,10	20,57	-48,43	20,78	-48,22	0,03	-68,97
2	69	26,53	-42,47	15,83	-53,17	33,67	-35,33	0,72	-68,28
3	68	4,38	-63,62	20,64	-47,36	20,91	-47,09	16,56	-51,44
4	37	0,28	-36,72	23,31	-13,69	17,84	-19,16	32,39	-4,61
5	37	0,98	-36,02	23,06	-13,94	17,77	-19,23	0,01	-36,99
6	25	0,28	-24,72	23,33	-1,67	16,94	-8,06	0,20	-24,80
7	69	2,65	-66,35	22,57	-46,43	17,86	-51,14	4,99	-64,01
8	68	13,76	-54,24	11,11	-56,89	26,17	-41,83	16,10	-51,90
9	68	13,97	-54,03	11,11	-56,89	24,51	-43,50	33,60	-34,40
10	68	6,34	-61,66	17,66	-50,35	18,04	-49,96	0,09	-67,91
11	68	6,79	-61,21	20,63	-47,38	17,99	-50,01	3,23	-64,77
12	68	1,24	-66,76	23,08	-44,92	13,39	-54,61	3,73	-64,27
13	69	6,85	-62,15	20,63	-48,38	16,69	-52,31	16,59	-52,41
14	69	14,23	-54,77	11,11	-57,89	21,02	-47,98	17,18	-51,82
15	69	42,01	-26,99	27,78	-41,22	32,09	-36,91	18,45	-50,55
16	69	0,78	-68,22	23,42	-45,58	4,32	-64,68	0,14	-68,86
17	69	19,72	-49,28	12,86	-56,14	15,06	-53,94	11,25	-57,75
18	69	15,01	-53,99	11,11	-57,89	13,25	-55,75	22,36	-46,64
19	69	15,49	-53,51	22,22	-46,78	13,19	-55,81	0,03	-68,97
20	31	0,20	-30,80	23,35	-7,65	2,20	-28,80	0,28	-30,72
21	31	0,15	-30,85	23,38	-7,62	2,14	-28,86	3,19	-27,81
22	69	3,12	-65,88	22,04	-46,96	3,82	-65,18	30,97	-38,03
23	69	10,70	-58,30	19,74	-49,26	9,90	-59,10	0,08	-68,92
24	69	17,45	-51,55	22,22	-46,78	12,35	-56,65	0,92	-68,08
25	69	17,90	-51,10	22,22	-46,78	11,90	-57,10	2,86	-66,14
26	21	0,01	-20,99	23,42	2,42	0,79	-20,21	13,97	-7,03
27	248	0,67	-247,33	23,16	-224,84	1,24	-246,76	14,56	-233,44
28	248	2,00	-246,00	22,03	-225,97	2,01	-245,99	15,95	-232,05
29	248	2,06	-245,94	22,01	-225,99	2,00	-246,00	17,34	-230,66
30	248	2,06	-245,94	23,40	-224,60	1,53	-246,47	17,95	-230,05
31	248	31,47	-216,53	17,50	-230,50	17,59	-230,41	0,03	-247,97
32	248	29,59	-218,41	15,83	-232,17	15,92	-232,08	0,17	-247,83
33	36	0,83	-35,17	22,92	-13,08	0,78	-35,22	4,96	-31,04
34	309	0,61	-308,39	23,42	-285,58	0,65	-308,35	16,07	-292,93
35	18	0,03	-17,97	23,41	5,41	0,07	-17,93	17,46	-0,54
36	309	0,67	-308,33	23,25	-285,75	0,62	-308,38	19,54	-289,46
37	16	0,03	-15,97	23,42	7,42	0,04	-15,96	0,01	-15,99
38	16	0,01	-15,99	23,42	7,42	0,01	-15,99	0,09	-15,91
39	16	0,09	-15,91	23,40	7,40	0,09	-15,91	11,21	-4,79
40	16	0,14	-15,86	23,44	7,44	0,14	-15,86	22,32	6,32
41	309	1,55	-307,45	23,30	-285,70	1,27	-307,73	0,25	-308,75
42	19	0,08	-18,92	23,41	4,41	0,08	-18,92	1,29	-17,71
43	309	0,70	-308,30	22,83	-286,17	0,61	-308,39	6,08	-302,92
44	19	0,03	-18,97	23,43	4,43	0,03	-18,97	8,37	-10,63
45	28	0,08	-27,92	23,43	-4,57	0,08	-27,92	17,01	-10,99
46	309	2,86	-306,14	22,65	-286,35	2,08	-306,92	17,60	-291,40
		$L_{m\acute{A}x}$	-15,86	$L_{m\acute{A}x}$	7,44	$L_{m\acute{A}x}$	-8,06	$L_{m\acute{A}x}$	6,32

APÊNDICE C –AMBIENTE 5 MÁQUINAS PARA REGRA LIFO ($L_{m\acute{A}x}$)

j	d_j	C_j	L_j
1	69	39,34	-29,66
2	69	49,21	-19,79
3	68	36,42	-31,58
4	37	33,38	-3,62
5	37	33,13	-3,87
6	25	32,29	7,29
7	69	33,29	-35,71
8	68	41,16	-26,84
9	68	40,55	-27,45
10	68	32,04	-35,96
11	68	31,34	-36,66
12	68	26,55	-41,45
13	69	30,05	-38,95
14	69	29,44	-39,56
15	69	44,91	-24,09
16	69	17,14	-51,86
17	69	27,25	-41,75
18	69	25,51	-43,49
19	69	25,26	-43,74
20	31	14,15	-16,85
21	31	13,96	-17,04
22	69	14,40	-54,60
23	69	12,11	-56,89
24	69	14,39	-54,61
25	69	13,82	-55,18
26	21	2,71	-18,29
27	248	3,28	-244,72
28	248	3,47	-244,53
29	248	2,69	-245,31
30	248	2,69	-245,31
31	248	18,33	-229,67
32	248	16,64	-231,36
33	36	1,31	-34,69
34	309	0,83	-308,17
35	18	0,25	-17,75
36	309	0,80	-308,20
37	16	0,21	-15,79
38	16	0,19	-15,82
39	16	0,17	-15,83
40	16	0,22	-15,78
41	309	1,30	-307,70
42	19	0,08	-18,92
43	309	0,61	-308,39
44	19	0,03	-18,97
45	28	0,08	-27,92
46	309	2,08	-306,92
$L_{m\acute{A}x}$			7,29