UNIVERSIDADE DE SÃO PAULO
FACULDADE DE ECONOMIA, ADMINISTRAÇÃO E CONTABILIDADE
DEPARTAMENTO DE ADMINISTRAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO

PROPOSTA DE UM MODELO EM PROGRAMAÇÃO LINEAR PARA A SOLUÇÃO DE PROBLEMAS DE SISTEMAS PRODUTIVOS *JOB SHOP* COM *SETUP* DEPENDENTES DA SEQUÊNCIA.

Alessandra Henriques Ferreira

Orientador: **Prof. Dr. Marcio Mattos Borges de Oliveira**

SÃO PAULO
2012

Prof. Dr. João Grandino Rodas
Reitor da Universidade de São Paulo

Prof. Dr. Reinaldo Guerreiro
Diretor da Faculdade de Economia, Administração e Contabilidade

Prof. Dr. Adalberto Américo Fischmann
Chefe do Departamento de Administração

Prof. Dr. Lindolfo Galvão Albuquerque
Coordenador do Programa de Pós-Graduação em Administração

ALESSANDRA HENRIQUES FERREIRA

PROPOSTA DE UM MODELO EM PROGRAMAÇÃO LINEAR PARA A SOLUÇÃO DE PROBLEMAS DE SISTEMAS PRODUTIVOS *JOB SHOP* COM *SETUP* DEPENDENTES DA SEQUÊNCIA.

Tese apresentada ao Departamento de Administração da Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo como requisito para a obtenção do título de Doutor em Administração.

Orientador: **Prof. Dr. Marcio Mattos Borges de Oliveira**

Versão Corrigida

(versão original disponível na Faculdade de Economia, Administração e Contabilidade)

SÃO PAULO

2012

FICHA CATALOGRÁFICA
Elaborada pela Seção de Processamento Técnico do SBD/FEA/USP

Ferreira, Alessandra Henriques
Proposta de um modelo em programação linear para a solução de problemas de sistemas produtivos job shop com setup dependentes da sequência / Alessandra Henriques Ferreira. -- São Paulo, 2012.
104 p.

Tese (Doutorado) – Universidade de São Paulo, 2012.
Orientador: Marcio Mattos Borges de Oliveira.

1. Programação linear 2. Pesquisa operacional 3. Scheduling
I. Universidade de São Paulo. Faculdade de Economia, Administração e Contabilidade. II. Título.

CDD – 003

AGRADECIMENTOS

Ao Prof. Dr. Marcio Mattos Borges de Oliveira, pela orientação, confiança, compreensão no desenvolvimento deste trabalho e amizade nesses anos.

A Prof. Dra. Sonia W. B. de Oliveira, pelas valiosas sugestões e incentivo para o desenvolvimento deste trabalho.

A todos os demais Professores que colaboraram para execução desta tese repassando seus conhecimentos.

Agradeço imensamente ao Edison pela ajuda, estímulo e compreensão ao longo destes anos de trabalho.

Agradeço aos meus pais, que sempre me deram o apoio necessário em todas as etapas de minha vida, pelo estímulo e compreensão ao longo destes anos.

Agradeço a todos que, direta ou indiretamente, colaboraram para a realização desta tese.

“Um especialista em resolver problemas deve ser dotado de duas qualidades incompatíveis – uma imaginação inquietante e uma paciente obstinação”.

Howard W. Eves

RESUMO

Problemas de sequenciamento são muito comuns, eles existem sempre que há uma escolha sobre a ordem em que várias tarefas podem ser realizadas. Seja o negócio uma companhia aérea, um hotel, um fabricante de computadores ou uma universidade, esses problemas fazem parte do cotidiano. A aplicação das técnicas de sequenciamento permite, por exemplo, a redução dos custos e o aumento na agilidade da cadeia de suprimentos, afetando as operações no inicio e no fim da cadeia de suprimentos pelo mundo inteiro. Este trabalho parte da intenção de abordar os princípios e as técnicas de Scheduling, com a finalidade de propor um modelo de sequenciamento para a solução de um problema em sistemas produtivos do tipo job shop com **n** tarefas e **m** máquinas, considerando os tempos de setup dependentes da sequência e tendo como horizonte de planejamento o curto prazo. O objetivo é o de minimizar a perda dos tempos não produtivos. Neste contexto, a pesquisa apresenta um enfoque tanto exploratório, quanto aplicado. Pode ser considerado exploratório, uma vez que a revisão da literatura é referência central para o desenvolvimento do modelo matemático. É aplicado considerando-se o desenvolvimento do modelo e avaliação de sua aplicabilidade. Sendo assim, a partir da definição do problema e desenvolvimento do modelo por meio do uso de técnicas matemáticas e abordagens da pesquisa operacional constatou-se que as conclusões tiradas podem inferir decisões para o problema real. Sendo que, as considerações aqui feitas têm por finalidade relatar os fatos constatados nos experimentos realizados, visando contribuir com futuras pesquisas na área.

Palavras-chave: planejamento e controle; *job shop*; *setup* dependente; *scheduling*.

ABSTRACT

Sequencing problems are very common, they happen every time there is a choice regarding the order in which several tasks can be performed. The business can be an airline, a hotel, a computer manufacturer or a university; these issues are part of their routine. The application of the sequencing techniques allows, for example, reducing the costs and fastening the supply chain all over the world. This work has an approach to Scheduling principles and techniques, with the objective of proposing a sequencing model for the solution of a problem in productive systems such as job shop with **n** tasks and **m** machines, considering setup times dependent on the sequence and adopting a short term planning. The goal is to minimize the waste of unproductive time. In this context, the research presents an approach both exploratory and applied. It can be considered exploratory, once that the literature review is a main reference to the development of a mathematical model. It is applied when we consider the development of the model and evaluation of its applicability. Thus, from the problem definition and the model development by the use of mathematical techniques and approaches of the operational research, we found that the conclusions drawn from the model might infer decisions for a real problem. The considerations shown here aim to report the facts given in the conducted experiments, intending to contribute to future researches in the area.

Keywords: planning and control; job shop; dependent setup; scheduling.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS **2**
LISTA DE QUADROS **3**
LISTA DE ILUSTRAÇÕES **4**
1 INTRODUÇÃO **5**
1.1 OBJETIVOS 6
1.2 JUSTIFICATIVA 6
1.3 ESTRUTURA DO TRABALHO 8
1.4 DELIMITAÇÕES DA PESQUISA 9
1.5 CONTRIBUIÇÃO DA TESE 9
2 REFERENCIAL TEÓRICO **10**
2.1 PLANEJAMENTO E CONTROLE DA PRODUÇÃO (PCP) 10
2.1.1 Previsão de demanda 23
2.1.2 Sistemas de coordenação de ordens de produção e compra (SCO) 34
2.2 CONTROLE DE CHÃO DE FABRICA: PROGRAMAÇÃO E *SCHEDULING* 35
2.2.1 *Scheduling* 35
2.2.2 *Scheduling*: estrutura e notação 38
2.2.3 *Job shop* com tempos de *setup* dependentes 44
2.2.4 Modelos de solução para problemas *Job shop Scheduling* 49
2.2.4.1 Métodos de Solução Ótima 49
2.2.4.1.1 Planos de Corte: Gomory 52
2.2.4.2 Métodos Heurísticos 57
3 O MÉTODO DE PESQUISA **61**
3.1 TIPO DE PESQUISA 62
3.2 COLETA DE DADOS: MÉTODOS E INSTRUMENTO 62
3.2.1 Tipos de dados 63
3.2.2 Técnicas de coleta e análise de dados 63
3.3 DESENVOLVIMENTO DO MODELO 64
3.4 ETAPAS DA PESQUISA 66
4 FORMULAÇÃO DO PROBLEMA **67**
5 MÉTODO DE RESOLUÇÃO **70**
6 RESULTADOS E DISCUSSÃO **74**
7 CONSIDERAÇÕES FINAIS **86**
7.1 RECOMENDAÇÕES PARA ESTUDOS FUTUROS NO TEMA 87
REFERÊNCIAS BIBLIOGRÁFICAS **88**
APÊNDICE **97**

LISTA DE ABREVIATURAS E SIGLAS

APO: Administração da Produção
ATO: Assembly to order
CNE: Controle do Nível de Estoque
CP: Controle da Produção
CPLEX: IBM ILOG CPLEX Optimizer
EDD: Earliest Due Date
ETO: Engineering to order
F: Flow Shop
FCFS: First Come, First Served
Fperm: Flow Shop Permutacional
G: Job Shop
GA: algoritmos genéticos
JSS: Job shop scheduling
JIT: Just-in-time
k-paralela: Máquinas paralelas
LPT: Longest Processing Time
LWKR: Least Work Remaining
MATLAB: MATrix LABoratory – Laboratório de matrizes
MOPNR: Most Operations Remainning
MPS: Master Production Schedule
MRP II: Manufacturing Resource Planning
MRP: Material Requirements Planning
MTO: Make to order
MTS: Make to stock
MWKR: Most Work Remaining
O: Open Shop
OPT: Optimized Production Technology
PCP: Planejamento e Controle da Produção
PI: Programação Inteira
PLI: Programação Linear Inteira
PMP: Programa Mestre de Produção
PO: Pesquisa Operacional
PP: Planejamento da Produção
QRTS: Quick response to stock
RTO: Resources to order
SCO: Sistema de Coordenação de Ordens
SDST JSS*:* Setup dependente para job shop scheduling
SFC: Shop Floor Control
SP: Sistema de Produção
SPT: Shortest Processing Time
TI: Tecnologia da Informação

LISTA DE QUADROS

Quadro 1 – Regras básicas entre os sistemas *job shop* e *flow shop* 19
Quadro 2 – Fatores, técnicas e abordagens que influenciam o PCP 23
Quadro 3 – Relação entre características importantes das previsões 24
Quadro 4 – Técnicas de previsão e modelos comuns 27
Quadro 5 – Níveis de planejamento e programação da produção 34
Quadro 6 – Principais notações de *scheduling* 39
Quadro 7 – Abreviações 42
Quadro 8 – Regras de prioridades 43
Quadro 9 – Subclasses de modelos de solução ótima 44
Quadro 10 – Literatura sobre *scheduling* em ambiente *job shop* com *setup* dependente 48
Quadro 11 – Dados do problema 1: tempos de produção (l_i) e respectivas demandas (d_i) 75
Quadro 12 – Os tempos de *setup* dependente do problema 1 75
Quadro 13 – Problema 1: dados da solução gerada 76
Quadro 14 – Dados do problema 2: tempos de produção (l_i) e respectivas demandas (d_i) 77
Quadro 15 – Os tempos de *setup* dependente do problema 2 77
Quadro 16 – Problema 2: dados da solução gerada 79
Quadro 17 – Dados do problema 3: tempos de produção (l_i) e respectivas demandas (d_i) 79
Quadro 18 – Os tempos de *setup* dependente do problema 3 80
Quadro 19 – Problema 3: dados da solução gerada 82
Quadro 20 – Tempos computacionais médios 83
Quadro 21 – Dados do problema 4: tempos de produção (l_i) e respectivas demandas (d_i) 84
Quadro 22 – Os tempos de *setup* dependente do problema 4 85

LISTA DE ILUSTRAÇÕES

Figura 1 – O processo de transformação 10
Figura 2 – Prazos, atividades e objetivos do sistema produtivos das organizações 11
Figura 3 – Equilíbrio entre as atividades de planejamento e controle 12
Figura 4 – A estrutura do PCP 13
Figura 5 – A estrutura do controle da produção 14
Figura 6 – Atividades do PCP 16
Figura 7 – Matriz produto-processo 17
Figura 8 – Características básicas do sistema produtivo 21
Figura 9 – Passos para o processo de previsão 25
Figura 10 – Fatores que compõem o ambiente externo e interno do planejamento agregado 30
Figura 11 – Estratégias de planejamento agregado 31
Figura 12 – Comportamento de alguns fatores x estratégia de atendimento a demanda 32
Figura 13 – Dados de entrada para a geração do MPS x estratégia de resposta à demanda 33
Figura 14 – Gráfico de Gantt 36
Figura 15 – Relação entre ambientes e máquinas 41
Figura 16 – Hierarquia de complexidade dos problemas de *scheduling* 45
Figura 17 – Classificação hierarquizada de modelos de solução ótima para o problema *job shop* 50
Figura 18 – Classificação hierarquizada dos modelos heurísticos de busca 58
Figura 19 – Classificação hierarquizada dos modelos heurísticos de passo único 60
Figura 20 – Processo de solução do problema 64
Figura 21 – Processo de produção 65
Figura 22 – Etapas da pesquisa 66
Figura 23 – Conceito de economia 72

1 INTRODUÇÃO

As grandes mudanças na programação da produção são devido a dois acontecimentos chave: a criação de Henry Gantt de caminhos úteis para entender as complexas relações entre homens, máquinas, ordens e o tempo; e o poder da tecnologia da informação (TI) para coletar, visualizar, processar, os dados com facilidade e rapidez auxiliando assim, os processos de tomada de decisão. Esses eventos levaram ao declínio dos controles manuais e ao crescimento do desenvolvimento de *softwares* e algoritmos de otimização para *scheduling* (HERRMANN, 2006).

Um ponto a ser observado é que muitas indústrias não se aproveitaram destas evoluções. Eles continuam a produzir bens para os seus clientes, mas o sequenciamento da produção ainda pode ser descrito como uma coleção quebrada de planos independentes que são frequentemente ignorados. Uma programação e sequenciamento efetivos da produção dependem de um planejamento efetivo (PALMER, 2006).

As organizações contemporâneas existem em um ambiente de grande agitação e dinamismo. As inovações tecnológicas e responsabilidades ambientais incentivam os gestores a desenvolver uma nova visão para as organizações. O aumento da concorrência levou à procura de novos métodos e processos de produção, com a intenção de melhoria da produtividade e atendimento de uma clientela mais informada e conhecedora do produto que deseja comprar (SOBRAL; PECI, 2008).

Os gerentes de produção precisam gerenciar as mudanças necessárias e melhorias na produção, desenvolvendo habilidades e ferramentas necessárias para obter sucesso na nova economia mundial. Devem buscar continuamente diferenciais competitivos em custos, qualidade, tempo de entrega ou flexibilidade. *Scheduling* é um processo de tomada de decisão que é usado em muitas indústrias, com o objetivo de alocar recursos e tarefas de acordo com os prazos e sua meta é otimizar uma ou mais medidas de desempenho.De acordo com Pacheco e Santoro (1999, p.2)

> os objetivos tratados nos problemas de *scheduling* podem ser resumidos no atendimento dos prazos (ou data de entrega), na minimização do tempo de fluxo ou dos estoques intermediários e na maximização da capacidade disponível, ou em uma combinação destes.

Scheduling é um processo de tomada de decisão que desempenha um papel importante na maioria das manufaturas e sistemas de produção (PINEDO, 2008). Parte dos problemas

estudados aplica-se ao ambiente *job shop*, que se caracteriza por ser um ambiente de fluxo flexível, de difícil programação devido à variabilidade dos itinerários das tarefas e introdução contínua de novas tarefas a serem processadas.

Este trabalho tem a intenção de estudar um problema de *scheduling* em ambiente *job shop* (produção de uma grande variedade de produtos) com **n** tarefas e **m** máquinas, considerando os tempos de *setup* dependentes da sequência e tendo como horizonte de planejamento o curto prazo. Segundo Fernandes e Godinho Filho (2010) existe muita controvérsia na literatura a respeito dos horizontes de planejamento, eles consideram para o curto prazo um período de até três meses, este trabalho terá como unidade de medida **horas** (por exemplo: 12 horas, 24 horas, 36 horas etc.).

1.1 OBJETIVOS

Com a finalidade de estudar o ambiente produtivo descrito acima estabeleceu-se como **objetivo geral**:

- Identificar e analisar um método que possibilite sequenciar um sistema *job shop* de forma a produzir o máximo possível (de acordo com a demanda) dentro de um horizonte de planejamento considerado de curto prazo respeitando a relação de dependência entre *setups*.

E como **objetivos específicos** este trabalho propõe um modelo de sequenciamento que:

- Minimize as perdas de tempos não produtivos (*setup* e folga).
- Atenda a demanda no curto prazo.

1.2 JUSTIFICATIVA

Problemas de sequenciamento são muito comuns, eles existem sempre que há uma escolha sobre a ordem em que várias tarefas podem ser realizadas. Seja o negócio uma companhia aérea, um hotel, um fabricante de computadores ou uma universidade, esses problemas fazem parte do cotidiano. A aplicação das técnicas de sequenciamento permite, por exemplo, a redução dos custos e o aumento na agilidade da cadeia de suprimentos, afetando as operações no inicio e no fim da cadeia de suprimentos pelo mundo inteiro. Hayes *et al.* (2008) afirmam que diante de fenômenos como a globalização e internacionalização da economia, percebe-se um acréscimo da competitividade nas organizações, que passaram a buscar melhorias contínuas nos seus negócios em geral e nos sistemas produtivos em particular.

Os primeiros desenvolvimentos no campo do sequenciamento foram motivados por problemas industriais, tanto que é natural usar-se o vocabulário de manufatura ao descrever problemas relacionados ao campo em questão. A teoria de *scheduling* é composta por uma série de técnicas para a resolução de problemas, e se tornou um ponto central para o desenvolvimento, aplicação, e avaliação de procedimentos de simulação, métodos de rede e soluções heurísticas. Baker (1974) afirma que a seleção de uma técnica apropriada depende da complexidade do problema, da natureza do modelo e da escolha de um critério como também de outros fatores; em muitos casos é mais apropriado considerar como alternativa um conjunto de técnicas. Por essa razão, a teoria de *scheduling* é uma área que trata tanto do estudo de metodologias como do estudo de modelos.

Os recentes avanços computacionais unidos a necessidade das indústrias tornarem seus processos produtivos mais eficientes, estimulou as pesquisas de modelos de otimização para o controle e planejamento dos sistemas produtivos. Levando em consideração o objetivo geral deste trabalho, pode-se avaliar com base na literatura a abordagem proposta de utilização do algoritmo de Gilmore e Gomory (1961) aplicado a um ambiente *job shop* respeitando a relação de dependência entre *setups*, como forma de minimizar as perdas dos tempos não produtivos como inovadora.

Em análise realizada no *site Web of Science*, utilizando a pesquisa de citações (*cited reference search*) dos artigos publicados por Ralph E. Gomory, verificou-se que 2564 artigos citaram publicações desse pesquisador entre 1958 e 2012. Desse total, 84 discorrem sobre *setup*, sendo que 10 consideram os tempos de *setup* dependentes, três sobre *flow shop* e nenhum deles trata isoladamente sobre *job shop*. Não foi encontrada na busca uma publicação que abordasse o conjunto de temas: cortes de Gomory, *setup* dependente e *job shop*.

Em relação ao *job shop scheduling* a literatura relata vários artigos começando por Jackson (1956), depois a técnica de *branch-and-bound* foi aplicada para minimizar o *makespan* em sistemas *job shop* (LOMNICKI, 1965; BROWN; LOMNICKI, 1966; BAKER; McMAHON, 1985; CARLIER; PINSON, 1989); as heurísticas para trabalhar com gargalos em sistemas *job shop* começaram com Adams, Balas e Zawack (1988) usando o algoritmo desenvolvido por Carlier (1982 *apud* PINEDO, 2008). Pacheco e Santoro (1999) propuseram uma proposta de classificação hierarquizada para problema *job shop*.

Devido à grande aplicação em indústrias os problemas de corte são clássicos na pesquisa operacional (PO), encontram-se artigos publicados em *journals* e revistas (MORABITO, 1994; MARQUES; ARENALES, 2002; LETCHFORD; LODI, 2002; KÖPPE *et al.*, 2004; GLOVER; SHERALI, 2005; POLDI; ARENALES, 2006; PILEGGI;

MORABITO; ARENALES, 2007; RANGEL; FIGUEIREDO, 2008; ZAMBELLI, 2009; POLDI; ARENALES, 2010; ESPINOZA, 2010).

Os sistemas de produção atuais são orientados ao mercado, sendo o cliente a força direcionadora dos esforços produtivos, sendo assim as organizações sabem que devem continuar buscando diferenciais competitivos: custos, qualidade, **tempo de entrega** ou flexibilidade. Segundo Fernandes e Godinho Filho (2010, p. 191) "a manufatura moderna geralmente requer distinção e ou diversificação de produtos, produção em pequenos lotes, rápida entrega e aderência a datas de entrega prometidas."

Considerando-se que um dos indicadores do nível de atendimento ao cliente é a resposta à demanda, espera-se que este trabalho contribua para um melhor entendimento da teoria de *scheduling*, inclusive como ferramenta que permita as organizações criarem diferenciais competitivos, como também identificar um novo modelo que permita a redução de perdas (*setup* e folgas), otimizando dessa forma o processo produtivo.

1.3 ESTRUTURA DO TRABALHO

Este trabalho é dividido em sete capítulos. O primeiro, já apresentado, refere-se à introdução e a uma visão geral do estudo. Descreve ainda os objetivos geral e específicos, bem como a justificativa, as delimitações definidas para o estudo.

O segundo capítulo trata da fundamentação teórica, através da qual é ressaltada a importância do tema do estudo; são relatados os principais conceitos e métodos que se tornam a fundamentação para o presente trabalho.

O terceiro capítulo descreve a metodologia utilizada, com definição do método de pesquisa, as estratégias de coleta e tratamento dos dados, bem como as delimitações próprias da pesquisa.

O quarto capítulo descreve a formulação do problema de sequenciamento em ambiente *job shop*, levando em consideração um horizonte de planejamento finito (curto prazo) e tempos de *setup* dependente.

No quinto capítulo apresenta-se o passo a passo para a resolução do problema formulado no capítulo anterior.

No sexto capítulo são discutidos os resultados obtidos pelo trabalho, apresentando o sequenciamento das linhas e suas análises.

O sétimo capítulo trata das considerações finais da pesquisa realizada e das recomendações para futuros trabalhos que envolvam a temática *scheduling* na área industrial.

Ao final encontram-se as referências bibliográficas utilizadas no decorrer do estudo e o apêndice contendo o código fonte das rotinas desenvolvidas em MATLAB complementares à pesquisa.

1.4 DELIMITAÇÕES DA PESQUISA

Diante da problemática a ser tratada, alguns conceitos tornam-se mais pertinentes de serem desenvolvidos, para sustentar a construção proposta o que, dada a abrangência, necessita que sejam estabelecidas as devidas delimitações onde se restringe esta pesquisa.

Estudou-se o algoritmo de corte de Gilmore e Gomory (1961) aplicado a ambientes *job shop*, com a finalidade de determinar a programação das **n** linhas de forma a atender a demanda do mercado no curto prazo, com o objetivo de minimizar as perdas de tempos não produtivos (*setup e* folga).

Trabalhou-se um subproblema, ou seja, um recorte do problema maior, onde linhas específicas foram sequenciadas respeitando os objetivos indicados.

Esta tese não tem como objetivo abordar a questão do último produto feito no horizonte de tempo anterior, como se tratam de várias linhas esses *setups* podem ser diferentes, então deixou-se a análise dessa questão para estudos posteriores.

1.5 CONTRIBUIÇÃO DA TESE

Dado um ambiente produtivo *job shop* com tempos de s*etup* dependente, uma das contribuições desta tese reside na definição do problema de sequenciamento nesse ambiente, levando em consideração um horizonte de planejamento finito (curto prazo). O desenvolvimento do modelo para a resolução desse problema também se configura como uma contribuição vista a ausência de publicações que abordem os temas planos de corte de Gomory, *setup* dependente e *job shop* de forma conjunta.

2 REFERENCIAL TEÓRICO

Na presente seção deste trabalho está delineada uma revisão conceitual. Nesse sentido, três grandes temas serão discutidos. De acordo com Baker (1974) as funções de planejamento e *scheduling* não são independentes, então em um primeiro momento descreve-se o planejamento e controle da produção e suas atividades. Em seguida, discute-se o porquê da programação, sequenciamento e controle das operações e teoria de *scheduling*. Finalizando com *job shop* e *setup* dependente: conceitos, métodos e aplicações.

2.1 Planejamento e Controle da Produção (PCP)

A maneira como são administrados os recursos produtivos é fundamental para o crescimento estratégico e para a competitividade das organizações. A administração da produção é a administração desses recursos. Incorpora o projeto e o controle de sistemas responsáveis pelo uso produtivo das matérias-primas, recursos humanos, equipamentos e instalações para o desenvolvimento de um produto ou serviço (CHASE; JACOBS; AQUILANO, 2006).

Na definição de Gaither e Frazier (2005, p. 5) “Administração da produção e operações (APO) é a administração do sistema da produção de uma organização, que transforma os insumos nos produtos e serviços da organização”, a Figura 1representa o modelo geral do processo de transformação da APO.

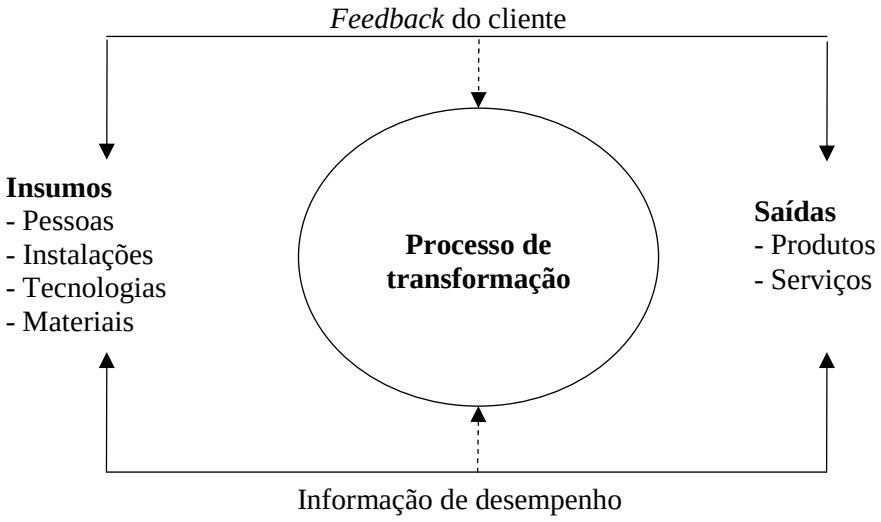

Figura 1 - O processo de transformação
Fonte: adaptado de Russell e Taylor III (2006)

Slack, Chambers e Johnston (2009) descrevem a administração da produção como: **importante, interessante, desafiadora**. Promovendo a criatividade que permite às empresas responderem a tantas mudanças. Assim, ela se preocupa com a administração de todo o sistema que produza um bem ou distribua um produto.

O sistema de produção (SP) é eficaz se os objetivos são de fato atingidos e é eficiente se os recursos são utilizados da melhor forma possível, ou seja, sem desperdícios. Além disso, o SP é efetivo se for simultaneamente eficaz e eficiente (FERNANDES; GODINHO FILHO, 2010).

De acordo com Tubino (2008) pode-se dividir o horizonte de planejamento de um sistema produtivo em três níveis: o longo, o médio e o curto prazo. A Figura 2 apresenta como esses prazos estão relacionados às atividades estratégicas, táticas e operacionais das organizações e os objetivos esperados com a realização dessas atividades.

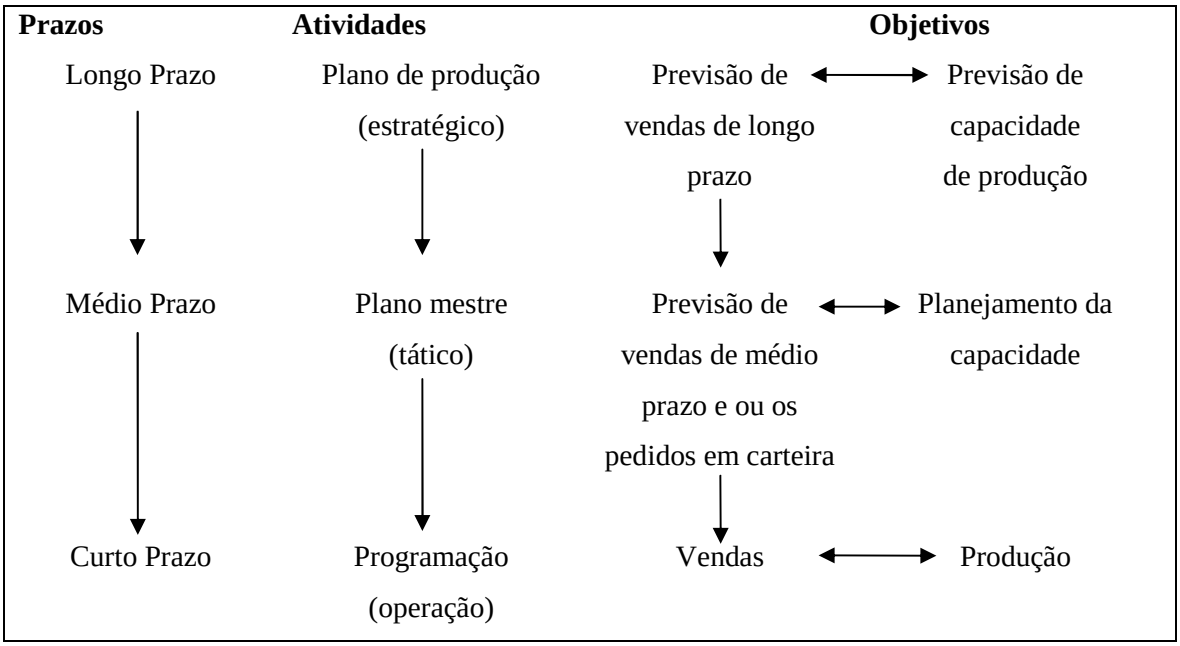

Figura 2 - Prazos, atividades e objetivos do sistema produtivos das organizações
Fonte: adaptado de Tubino (2008)

As questões estratégicas normalmente são amplas e atuam no longo prazo, causam mais impacto na eficácia de longo alcance da organização em termos de como esta pode direcionar as necessidades de seus clientes. Para se obter sucesso essas estratégias precisam estar alinhadas com a estratégia coorporativa. Nessa fase precisa-se montar um **plano de produção**, cuja função é considerar com que capacidade de produção o sistema deverá trabalhar de forma a atender os seus clientes (internos e externos), o plano deverá basear-se na previsão de vendas do longo prazo.

As decisões táticas atuam no médio prazo direcionam eficientemente a programação

dos insumos dentro das restrições estratégicas estabelecidas anteriormente. O **plano mestre de produção** busca atender às previsões de venda de médio prazo e ou os pedidos em carteira já negociados com os clientes. É chamado de tático porque deve analisar diferentes formas de manobrar o sistema produtivo disponível.

Uma vez organizado o sistema produtivo estruturado e a tática de operação definida, deve-se executar a **programação da produção** de forma a produzir bens e ou serviços e entregá-los aos clientes. Essa é uma atividade de curto prazo (escala de trabalho, prioridades de produção, entre outras situações que estão fortemente vinculadas ao chão de fábrica), neste nível só resta operar dentro de uma tática já planejada. Alterações de táticas no curto prazo geralmente geram desencontros entre os diferentes setores produtivos, pois não existe tempo hábil para sincronizar o processo como um todo. A formação de estoques é um dos possíveis resultados desses desencontros (FERNANDES; GODINHO FILHO, 2010; TUBINO, 2008; MARTINS; LAUGENI, 2005).

De acordo com Slack, Chambers e Johnston (2009) intervenções de curto prazo são possíveis se os acontecimentos não ocorrerem de acordo com o planejamento. Planos contingenciais deverão existir de forma que permitam pequenos desvios dos programas iniciais, eles agirão como recurso **reserva** e tornarão o PCP mais fácil no curto prazo. A Figura 3 ilustra como os aspectos do PCP variam no longo, médio e curto prazo.

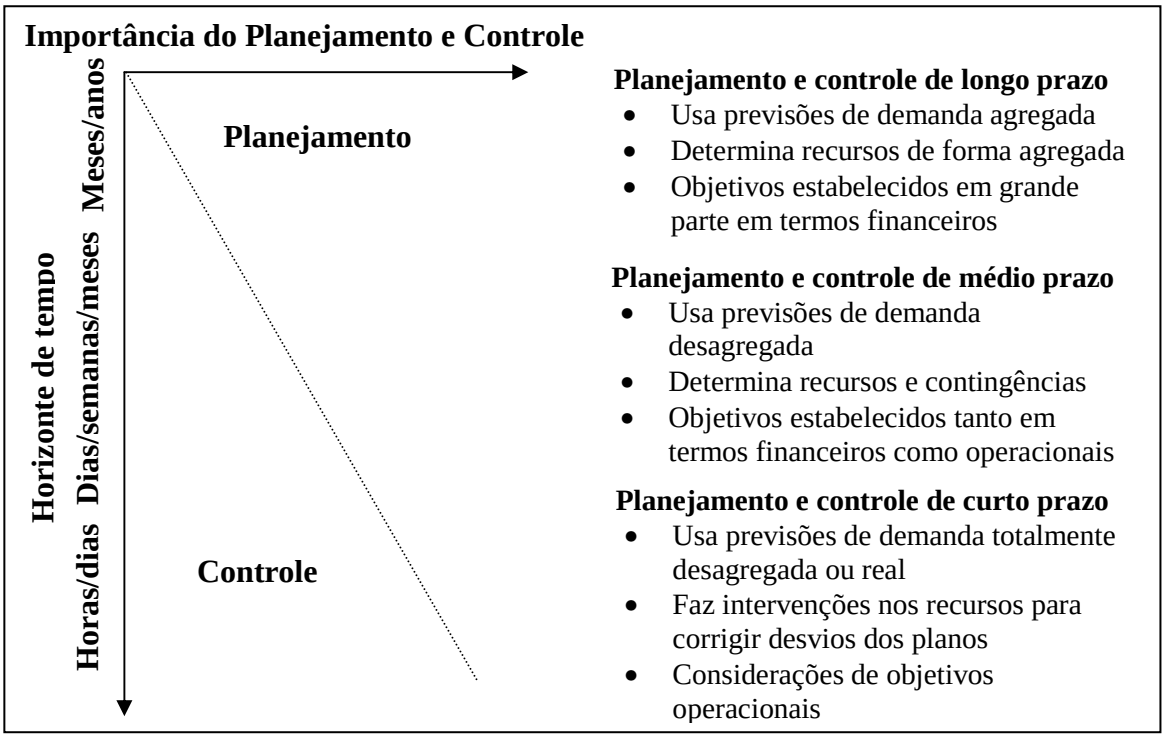

Figura 3 - Equilíbrio entre as atividades de planejamento e controle
Fonte: adaptado de Slack, Chambers e Johnston (2009)

Segundo Tubino (2008, p. 2) “um sistema produtivo será tão mais eficiente quanto consiga sincronizar a passagem de estratégias para táticas e de táticas para operações de produção e venda dos produtos solicitados.”

O PCP é o responsável por organizar os dados e tomar as decisões em relação as atividades escalonadas no tempo, ou seja, ele deve coordenar e aplicar os recursos produtivos de forma a atender da melhor maneira possível aos planos estabelecidos nos níveis estratégico, tático e operacional (MARTINS; LAUGENI, 2005; TUBINO, 2008). As decisões do PCP seguem uma estrutura hierárquica que é apresentada na Figura 4.

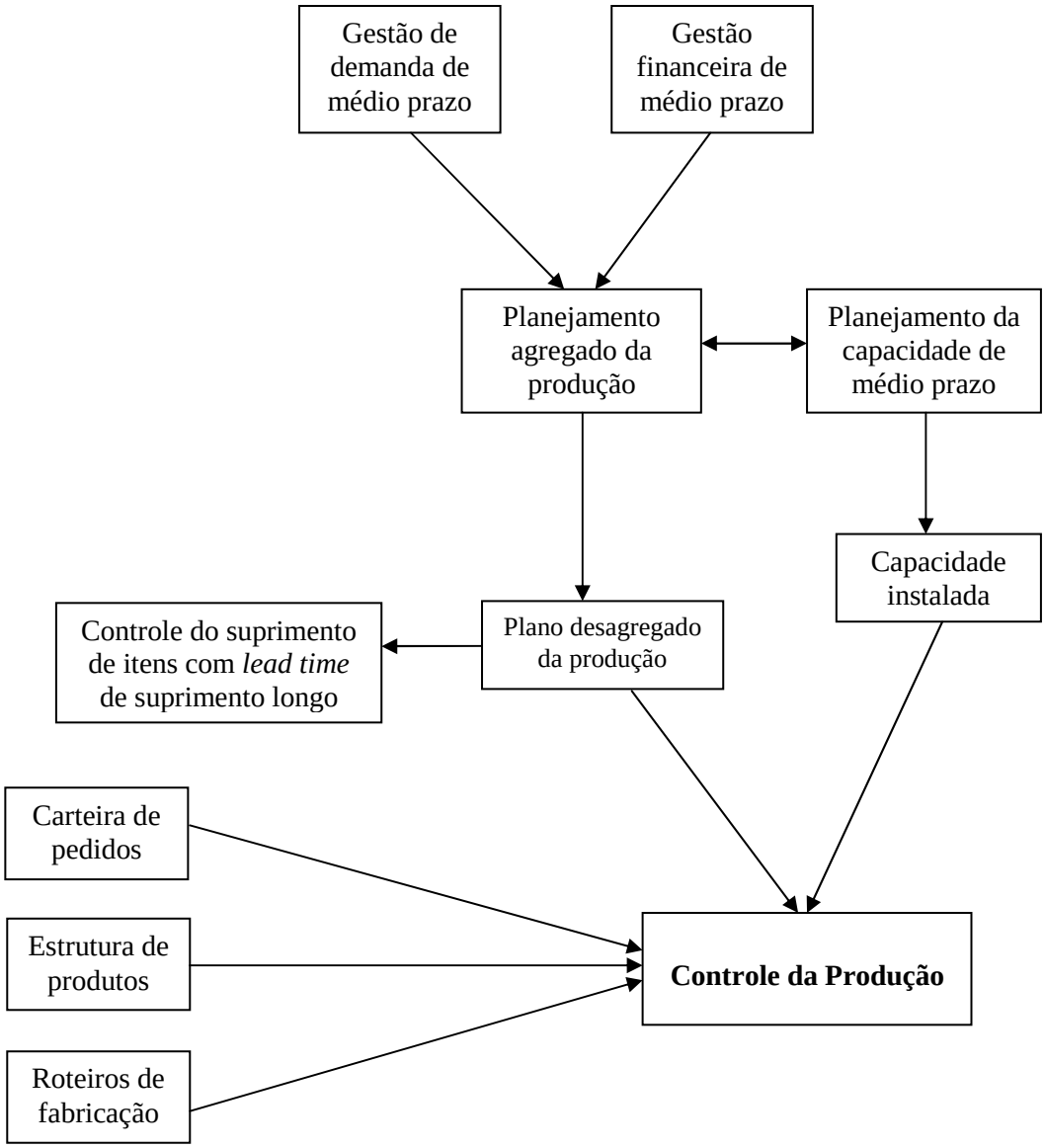

Figura 4 - A estrutura do PCP
Fonte: adaptado de Fernandes e Godinho Filho (2010)

Retomando a Figura 4, o planejamento da produção começa com a gestão da demanda no médio prazo, que tem como meta conhecer a demanda por meio das previsões. A previsão

de demanda juntamente com a gestão financeira de médio prazo são *inputs* fundamentais para a realização do planejamento agregado da produção. O planejamento de capacidade também é importante, pois interage com as decisões de planejamento agregado. O planejamento da Produção (PP) relaciona-se as atividades de médio prazo, em geral entre três e 18 meses, e assim toma decisões, na forma agregada, em termos de:

- **O que** produzir, comprar e entregar;
- **Quanto** produzir, comprar e entregar;
- **Quando** produzir, comprar e entregar;
- **Quem** e/ou onde e/ou como produzir.

O controle da produção (CP) é uma atividade gerencial e é responsável por regular, no curto prazo, geralmente até três meses, o fluxo de materiais de um sistema de produção. O CP faz os ajustes que permitem que a operação atinja os objetivos que o plano estabeleceu, mesmo que os pressupostos assumidos pelo plano não se confirmem (SLACK; CHAMBERS; JOHNSTON, 2009). A Figura 5 apresenta a estrutura do controle da produção.

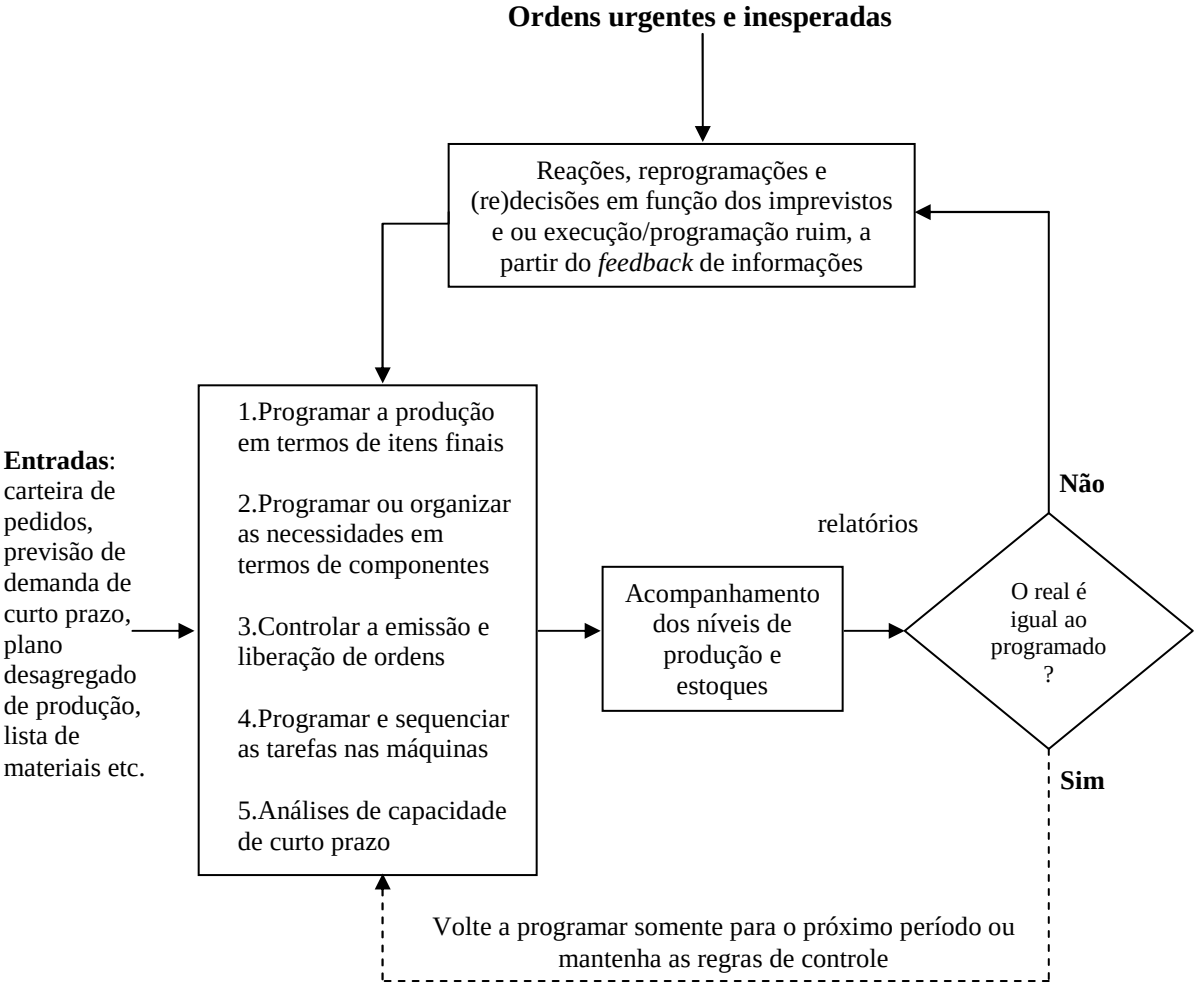

Figura 5 - A estrutura do controle da produção
Fonte: adaptado de Fernandes e Godinho Filho (2010)

A partir das apresentações das estruturas de Planejamento e Controle da Produção pode-se identificar resumidamente quais as principais atividades do PCP. Segundo Fernandes e Godinho Filho (2010) são:

- Prever a demanda;
- Desenvolver um plano de produção agregado;
- Realizar um planejamento da capacidade que suporte o planejamento agregado;
- Desagregar o plano agregado;
- Programar a produção no curto prazo em termos de itens finais (programa mestre de produção);
- Analisar a capacidade no nível do programa mestre de produção (PMP);
- Controlar por meio de regras de controle ou programar as necessidades em termos de componentes e materiais e avaliar a capacidade no nível do sistema de coordenação de ordens (SCO);
- Controlar a emissão e liberação das ordens de produção e compra, determinando se e quando liberar as ordens;
- Controlar estoques; e
- Programar e sequenciar as tarefas nas máquinas (*scheduling*).

Das atividades listadas acima, as quatro primeiras estão vinculadas ao PP e as outras cinco ao CP. Além dessas existem outras atividades específicas que também fazem parte do PCP, como: escolher e implantar um conjunto de princípios para controlar o fluxo de materiais; balancear as linhas; coordenar projetos; rearranjar instalações produtivas; estruturar as decisões do PCP com a estratégia de produção adotada integrando-as com outras áreas da empresa, entre outras.

Slack, Chambers e Johnston (2009) identificam quatro atividades superpostas: carregamento, sequenciamento, programação e controle:

Carregamento: é a quantidade de trabalho alocado para um centro de trabalho. Existindo o carregamento finito (quando se aloca trabalho até um limite estabelecido) e infinito (não limita a aceitação do trabalho, em vez disso tenta-se corresponder a ele).

Sequenciamento: seja a abordagem do carregamento finito ou infinito, quando o trabalho chega, decisões devem ser tomadas sobre a ordem em que as tarefas serão executadas.

Programação: uma vez determinada a sequência de execução das tarefas, as operações precisam de um cronograma detalhado, com a indicação do momento em que os trabalhos deverão começar e finalizar.

Controle: tendo criado um plano para a operação por meio do carregamento, sequenciamento e programação, cada etapa da operação precisa ser monitorada de forma a garantir que o planejado esteja de fato ocorrendo.

A Figura 6 mostra essas atividades e sua sobreposição.

Figura 6 - Atividades do PCP
Fonte: adaptado de Slack, Chambers e Johnston (2009)

O grau de complexidade de cada uma das atividades desenvolvidas pelo PCP dependerá do tipo de sistema produtivo. Segundo Moreira (2004) a classificação dos sistemas de produção é de grande utilidade na classificação das técnicas de planejamento e gestão da produção. Tradicionalmente, os sistemas produtivos são agrupados em três grandes categorias:

- Produção contínua ou de fluxo em linha;
- Produção por lotes ou por encomenda (fluxo intermitente); e
- Produção de grandes projetos sem repetição.

Slack, Chambers e Johnston (2009) afirmam que a natureza geral de qualquer processo é fortemente influenciada pelo volume e variedade que é processado, e o conceito desses tipos de processos sintetiza como essa relação (volume-variedade) afeta o projeto geral do processo. Vários autores subdividem as três categorias tradicionais dos sistemas produtivos, em ordem de aumento de volume e diminuição de variedade: por tarefa (*jobbing*) ou projetos, lote ou batelada, produção em massa e processos contínuos (KRAJEWSKI; RITZMAN; MALHOTRA, 2009; SLACK; CHAMBERS; JOHNSTON, 2009; RUSSELL; TAYLOR III, 2006; CORRÊA; CORRÊA, 2007). A Figura 7 ilustra as posições dos tipos de processos na matriz produto.

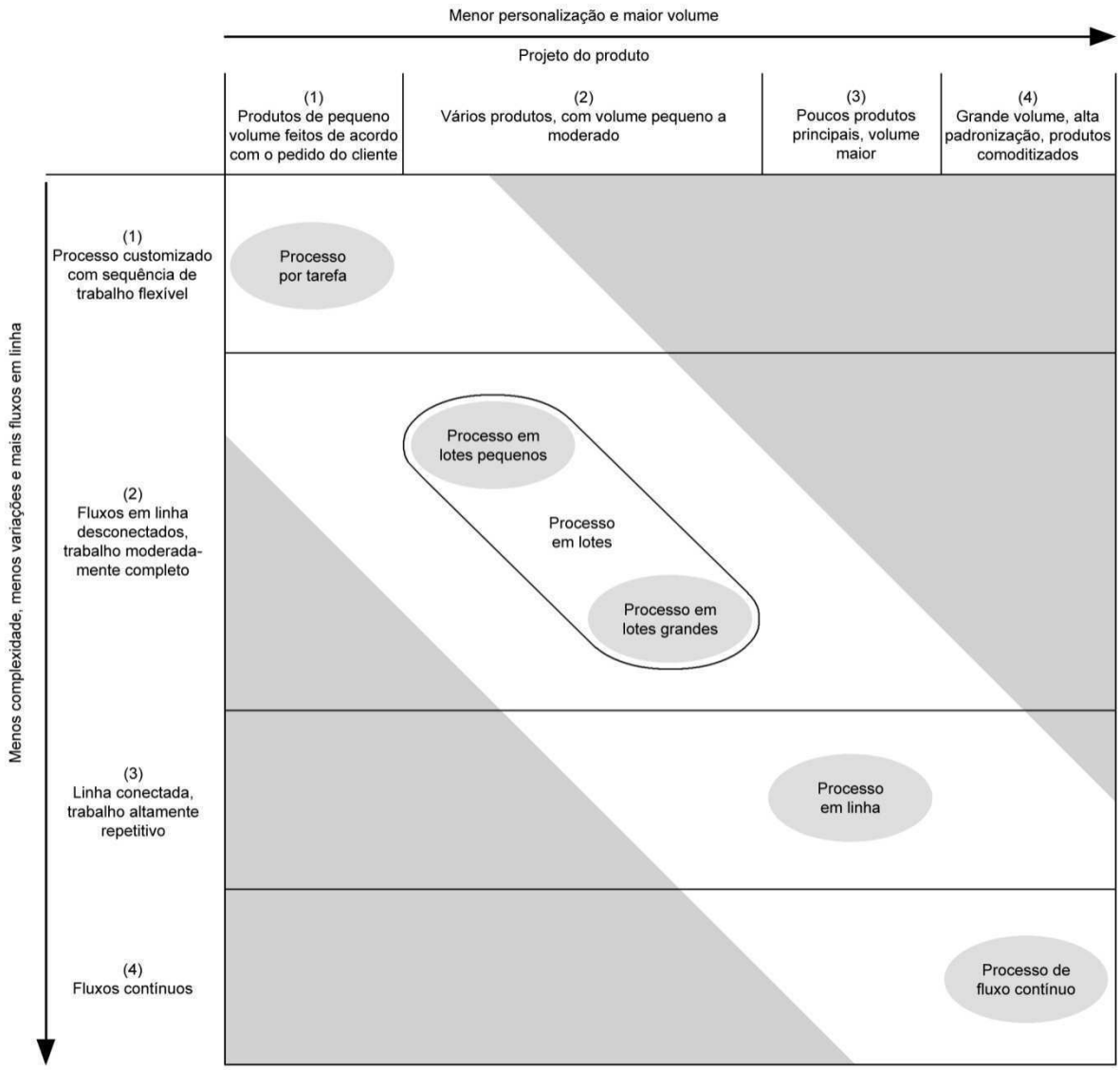

Figura 7 - Matriz produto-processo
Fonte: Krajewski, Ritzman e Malhotra (2009, p.108)

O sistema de **produção contínua**, ou de fluxo contínuo, é muito utilizado por empresas que produzem alto volume de produto, sem modificações por um longo período de tempo, ou seja, muito padronizados. O ritmo de produção é acelerado e as operações são executadas sem interrupção ou mudança, sendo o sistema em questão altamente automatizado. A produção contínua coloca cada processo produtivo em sequência linear para que o material de produção se movimente de uma máquina para outra continuamente e, para que, quando completado, seja transportado para a montagem do produto final. Isso é feito, muito em função do grande volume de produtos que precisam ser gerados (RUSSELL; TAYLOR III, 2006; SLACK; CHAMBERS; JOHNSTON, 2009).

O plano de produção é feito antecipadamente e pode cobrir um mês ou um ano, explorando ao máximo as possibilidades de recursos da empresa e proporcionando condições ideais de eficiência e eficácia.

Pode-se identificar como sendo as principais características do sistema de produção contínua:

- O produto é mantido em fabricação durante um longo período de tempo sem modificação, sendo rigidamente especificado em suas características e o processo produtivo é estabelecido em detalhes o que permite planejar em longo prazo todos os materiais e mão de obra necessários.
- A produção contínua facilita o detalhamento do planejamento.
- Como a produção é de longo prazo e destinada a enormes quantidades de produto é possível prever despesas e investimentos em equipamentos, moldes, ferramentas, proporcionando economia nos custos de produção.
- A produção contínua facilita as ações corretivas para resolver rapidamente qualquer problema de paralisação no processo de produção, seja por falta de material, por manutenção de máquina ou por falta de mão de obra. Ela facilita também a verificação diária do rendimento de produção em todos os pontos do processo produtivo, bem como permite que se faça um inventário regular dos materiais em processamento ou disponíveis em estoque no almoxarifado.
- O sucesso desse sistema de produção consiste totalmente no planejamento detalhado que deve ser feito antes de se iniciar a produção de um novo produto.

A **produção em massa** é um tipo de produção de fluxo contínuo. Segundo Russell e Taylor III (2006) a produção em massa produz grandes volumes de um produto padrão para um mercado de massa. A demanda desses produtos é estável e o volume é alto. Os bens que são produzidos em massa incluem automóveis, televisões, computadores pessoais, *fast food*, entre outros. Segundo Slack, Chambers e Johnston (2009) a produção contínua situa-se um passo além da produção em massa, pelo fato de trabalhar com volumes ainda maiores e em geral possuírem menor variedade de produtos.

O **sistema de produção em lote ou batelada** é o tipo de produção das empresas que produzem uma quantidade limitada de um tipo de produto de cada vez. Cada lote de produção é dimensionado para atender um determinado volume de vendas previsto em um determinado período de tempo. Terminado um lote de produção, a empresa inicia imediatamente a

produção de outro lote, e assim por diante. Por essa característica, esse tipo de produção também é conhecido por produção **intermitente** (KRAJEWSKI; RITZMAN; MALHOTRA, 2009).

Johnson e Montgomery (1974 *apud* FERNANDES; GODINHO FILHO, 2010, p. 2) classificam os sistemas intermitentes em *flow shop e job shop*.

- Sistema intermitente *flow shop*: onde todos os itens produzidos numa linha têm a mesma sequência de operações nas diversas máquinas.
- Sistema intermitente *job shop*: onde os itens fabricados em um setor produtivo não têm o mesmo roteiro de fabricação.

O Quadro 1 apresenta um resumo das diferenças básicas entre esses dois sistemas de produção com fluxo intermitente.

Quadro 1 - Regras básicas entre os sistemas *job shop* e *flow shop*

JOB SHOP	***FLOW SHOP***
Opera em lotes.	Opera em um fluxo de materiais e peças
Varia a produção variando o tamanho dos lotes ou frequência dos lotes.	Varia a produção alterando a taxa de produção.
Tende a ter custos maiores de *setup*.	Tende a ter custos menores de *setup*.
Materiais são trazidos para os departamentos ou centros de trabalho onde cada operação é realizada. Filas nos centros de trabalho são maiores.	As operações de tipos diferentes são sequenciadas de modo que o fluxo seja mantido. Filas são pequenas e variações têm que ser acompanhadas.
Utilização de equipamentos de uso geral.	Utilização de equipamentos de uso especializado (dedicado).

Fonte: adaptado de Fernandes e Godinho Filho (2010)

O sistema de produção intermitente (em lote ou batelada) apresenta as seguintes características:

- A fábrica é capaz de produzir produtos com diferentes características.
- As máquinas são agrupadas em grupos de mesmo tipo: o trabalho passa de um grupo para o outro em lotes de produção intermitentes, sendo que cada grupo constitui um departamento ou seção produtiva.
- Para cada lote de um diferente produto, as máquinas e ferramentas devem ser modificadas, adaptadas e arranjadas para atender aos diferentes produtos.
- A produção em lotes permite uma utilização regular da mão de obra sem grandes picos de produção.

- A produção em lotes exige grandes áreas de estocagem de produtos acabados e de matéria prima.
- Impõe a necessidade de um plano de produção bem feito e que possa integrar novos lotes de produção à medida que outros sejam completados, ou seja, aqui o plano de produção deve ser constantemente replanejado e atualizado, pois o sucesso do processo produtivo depende diretamente do plano de produção.

Já o **sistema de produção sob encomenda**, que é composto pelos processos de projeto e por tarefa (*jobbing*) é o sistema de produção utilizado pela empresa que produz somente após ter recebido o pedido ou a encomenda de seus produtos. Apenas após o contrato ou encomenda de um determinado produto é que a empresa vai produzi-lo para o cliente. **Os processos de projeto** geram produtos altamente customizados (baixo volume e alta variedade) sendo seu período de produção longo. **Os processos por tarefa** *(jobbing)* também lidam com alta variedade e baixo volume, mas diferentemente dos processos de projeto precisam compartilhar os recursos de operação com diversos outros processos, não há aqui a recursos dedicados com nos projetos (RUSSELL; TAYLOR III, 2006; SLACK; CHAMBERS; JOHNSTON, 2009).

O sistema de produção sob encomenda apresenta as seguintes características:

- Cada produto é único e específico: normalmente cada produto é único e de grande tamanho e complexidade, o que exige muito tempo para sua construção. O produto/serviço pode incluir ainda, características exclusivas solicitadas pelo cliente. Cada pedido ou contrato costuma ser considerado um produto específico, exigindo a identidade do produto ao longo de toda a sua fabricação.
- Cada produto exige uma variedade de máquinas e equipamentos: a construção do produto exige uma variedade de máquinas universais bem como uma oficina-base onde são construídas as partes daquilo que será o produto final.
- Cada produto exige uma variedade de operários especializados: sua construção exige uma série de operários especializados, capazes de executar cada uma das partes que compõe o produto final. Há uma demanda fluente de mão de obra especializada no local onde o trabalho será realizado.
- Cada produto exige uma data específica de entrega: há necessidade de se programar a entrega conforme os pedidos individuais, o que significa um compromisso de produção. As datas devem ser atendidas, fazendo com que o produto seja entregue ao cliente conforme os prazos sejam por ele solicitados.

- É difícil fazer previsões de produção, pois cada produto exige um trabalho complexo e demorado que é diferente dos demais produtos, por isso, cada um deles, exige um plano de produção específico.

Com base na descrição dos tipos de sistemas produtivos, pode-se afirmar que a classificação mais significativa para entender a complexidades das funções do PCP está relacionada com o grau de padronização dos produtos e consequente volume de produção demandado pelo mercado. A Figura 8 apresenta um resumo das características básicas de cada um dos sistemas produtivos.

Contínuos Massa	**Em lote ou batelada (intermitentes)**	**Sob encomenda**
Alta	Demanda/volume de produção	Baixa
Baixa	Flexibilidade/variedade de itens	Alta
Curto	*Lead time* produtivo	Longo
Baixos	Custos	Altos

Figura 8 - Características básicas do sistema produtivo
Fonte: adaptado de Tubino (2008)

De forma geral, os sistemas contínuos envolvem a produção de bens ou serviços que não podem ser identificados individualmente, e os sistemas em massa, em lotes e sob encomenda envolvem a produção de bens ou serviços que podem ser isolados, em lotes ou unidades. Vale ressaltar que a classificação dos sistemas produtivos não depende do tipo de produto em si, mas sim da forma como os sistemas são organizados para atender à demanda. Pode-se dizer que à medida que a demanda torna-se mais diversificada, e os lotes diminuem as funções de PCP ficam mais complexas (TUBINO, 2008).

Jonsson e Mattsson (2003) afirmam que o PCP pode ser influenciado por várias variáveis relacionadas ao produto, a demanda e o processo industrial respectivamente. Fernandes e Godinho Filho (2010) com base no trabalho desses autores organizaram os fatores que influenciam as atividades de PCP em três grupos: fatores relacionados ao produto, ao processo produtivo e ao ambiente externo.

As variáveis relacionadas ao produto que são consideradas críticas na perspectiva do PCP:

- **Estrutura complexa**: grande número de níveis na lista de materiais e um grande número de itens em cada nível.
- **Variedade de produtos:** influência no sistema de coordenação de ordens. O grau de variedade pode ser dividido em grau de distinção (muitos produtos semelhantes) e diversificação (muitos produtos diferentes).
- **Valor agregado do produto:** esse fator influencia muito nos métodos utilizados nas atividades de PCP, de forma geral itens com maior valor devem ser tratados com mais atenção.
- **Ciclo de vida dos produtos:** estágios diferentes do ciclo de vida de um produto podem requerer diferentes métodos de PCP.

As variáveis relacionadas aos fatores externos caracterizam demanda e fluxo de material. As variáveis seguintes são consideradas críticas:

- **Tipo de demanda:** a demanda poderá ser constante ao longo do período, irregular ou sazonal. Sua variação influencia na escolha do método de previsão a ser usado.
- **Estrutura do mercado:** tem forte impacto no planejamento agregado da produção, pois se refere às condições que o mercado oferece para a empresa em relação a fornecedores, mão de obra, entre outros.
- **Característica da demanda (previsibilidade e estabilidade):** se a demanda é dependente ou independente. Interfere nos métodos de previsão de demanda e nos sistemas de coordenação de ordens.

Alguns fatores relacionados ao processo produtivo que influenciam as atividades de PCP são os seguintes:

- ***Mix* de produtos:** a relação volume/variedade de produtos em um processo produtivo é um dos fatores que mais influenciam as atividades de PCP.
- ***Layout* das instalações:** o tipo de *layout* (funcional, por produto, celular, posição fixa) influencia as atividades de coordenação de ordens e programação de operações.
- **Tempos de *setup*:** o tempo de preparação da máquina para poder iniciar-se outra tarefa, influencia na escolha dos métodos a serem utilizados nas atividades de coordenação de ordens e programação da produção.
- **Tempo de fluxo:** a escolha de sistemas de coordenação de ordens tem relação com esse fator, que está ligado diretamente aos *lead times*. Esse é o tempo que o produto leva para percorrer todos os processos produtivos (incluindo tempo de filas).

Em resumo, pode-se afirmar que os fatores externos têm maior impacto nas atividades do Planejamento da Produção como, por exemplo, Planejamento Agregado. Já os fatores relacionados ao processo produtivo impactam mais nas atividades de Controle da Produção como, por exemplo, na Coordenação de Ordens e Programação da Produção. Por fim, os fatores relacionados ao produto têm um forte impacto tanto no Planejamento quanto no Controle da Produção. O Quadro 2 apresenta resumidamente outras questões importantes que influenciam o PCP.

Quadro 2 - Fatores, técnicas e abordagens que influenciam o PCP

Fatores que afetam as decisões no PCP	Os critérios competitivos da manufatura.
	Tipo de demanda.
	Tipo de produto.
	As características do processo produtivo.
	As características do fornecimento de recursos ao processo produtivo.
Principais técnicas utilizadas no PCP	Técnicas de previsão de demanda.
	Técnicas de planejamento da produção.
	Técnicas de programação da produção.
	Técnicas de controle da produção.
	Técnicas de gestão de estoques.
Abordagens do PCP	*Material Requirements Planning* (MRP/MRPII)
	Just-in-Time (JIT), *Optimized Production Technology* (OPT).
	Utilização de sistemas com capacidade finita.
	Abordagem de projetos.

Fonte: adaptado de Martins e Laugeni (2005)

Vale dizer que, no campo do PCP, a previsão também é um fator muito importante, uma vez que ela é um dos principais dados de entrada para funções e decisões do planejamento e controle da produção (FERNADES; GODINHO FILHO, 2010).

2.1.1 Previsão de demanda

Segundo Tubino (2008) as organizações direcionam suas atividades para o rumo em que elas acham que o seu negócio irá caminhar. Esse direcionamento normalmente é traçado

com base em previsões, sendo a principal delas a previsão de demanda.

A previsão é uma predição do que acontecerá no futuro. Metereologistas prevêem o tempo e as organizações tentam predizer quanto de seu produto será vendido no futuro. Uma previsão de demanda de produto é a base para a maioria das decisões de planejamento importantes (RUSSELL; TAYLOR III, 2006).

As previsões dentro do PCP podem ser classificadas de acordo com o horizonte de planejamento a que se destina: longo, médio e curto prazo. Segundo Fernandes e Godinho Filho (2010) as previsões são importantes:

- No longo prazo, para o planejamento de novas instalações, de novos produtos, gasto de capital, entre outros.
- No médio prazo, são a base para o planejamento agregado da produção e análise de capacidades agregadas.
- No curto prazo, ajudam na programação da força de trabalho, na programação de compras, nas análises de capacidade de curto prazo.

O Quadro 3 apresenta a relação entre algumas informações iniciais importantes, como por exemplo, o horizonte de planejamento da decisão necessária, o nível de agregação, o grau de detalhe requerido pela previsão (mensais ou semanais) e o nível de exatidão necessário. Outras informações devem ser também consideradas de forma a definir os esforços e recursos necessários à previsão: o que será previsto, o número de itens a serem previstos, o valor agregado dos itens, o volume dos recursos a serem utilizados (mão de obra, tempo computacional etc.).

Quadro 3 - Relação entre características importantes das previsões

Horizonte de planejamento	Nível de agregação	Grau de detalhes requerido	Nível de exatidão necessário
Longo prazo	Alto	Previsões mensais	Médio
Médio prazo	Médio	Previsões mensais ou semanais	Médio/Alto
Curto prazo	Baixo	Previsões semanais	Alto

Fonte: adaptado de Fernandes e Godinho Filho (2010)

Prever não significa simplesmente identificar e usar um método para calcular uma estimativa numérica do qual será a demanda no futuro. É um processo contínuo que exige monitoração e ajuste constante, a Figura 9 ilustra esse processo.

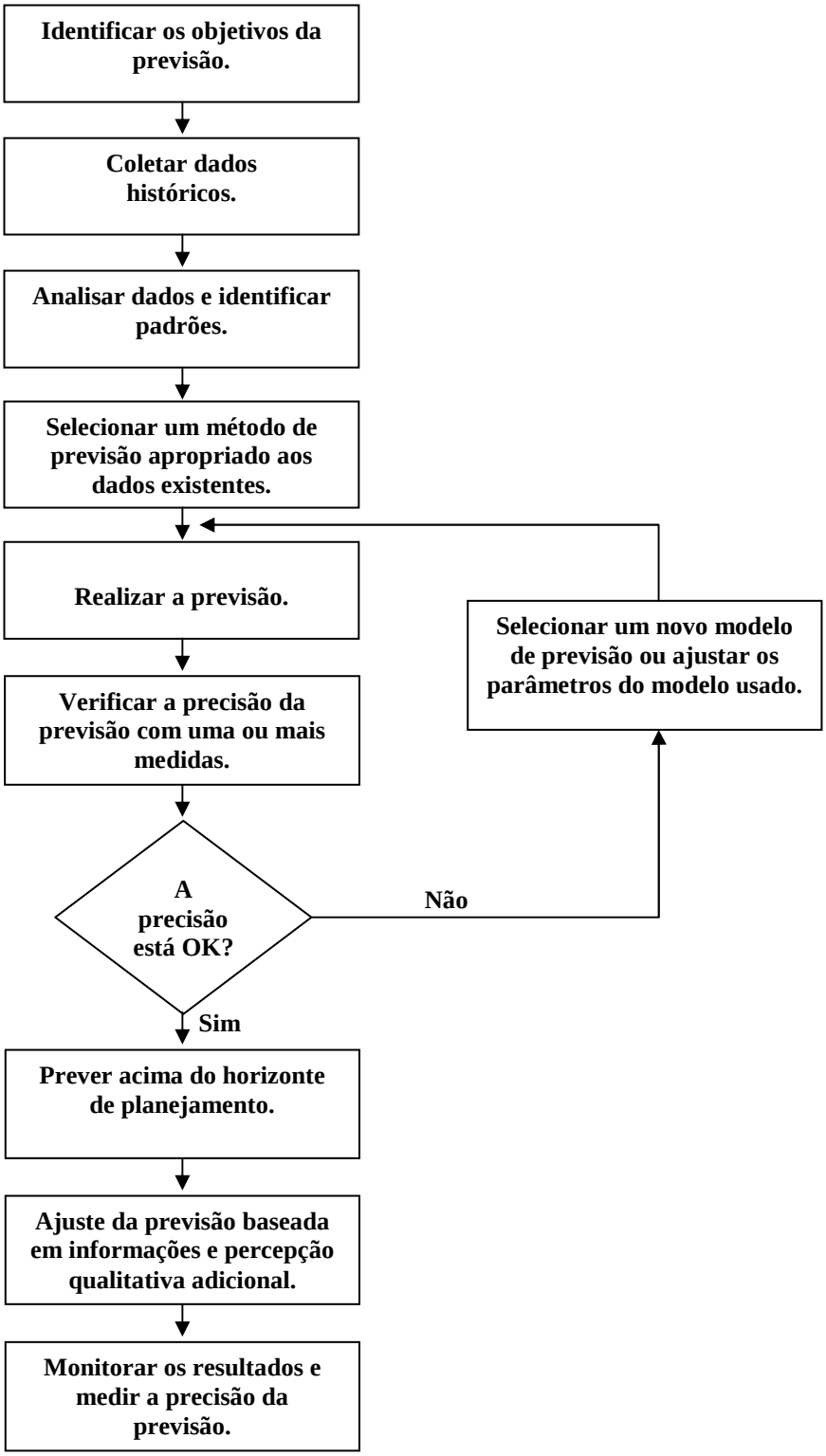

Figura 9 - Passos para o processo de previsão
Fonte: adaptado de Russell e Taylor III (2006)

Corrêa e Corrêa (2007, p. 250) afirmam que “a previsão, principalmente a de demanda, é, em geral, um dos assuntos mais controversos dentro das organizações e um dos que mais suscitam polêmica entre setores”. Esses mesmos autores apresentam os quatro principais erros cometidos pelas empresas quanto a previsões:

- **Erro 1:** confundir previsões com metas e, um erro subsequente, considerar as metas como se fossem previsões.
- **Erro 2:** gastar tempo discutindo se **acerta** ou **erra** nas previsões, quando o mais importante é discutir **o quanto** se está errando e como mudar os processos envolvidos, de forma a reduzir os erros.
- **Erro 3:** levar em conta, nas previsões que servirão para apoiar decisões em operações, um número só. As previsões dessa área devem sempre ser consideradas com dois números, a previsão em si e uma estimativa do erro dessa previsão.
- **Erro 4**: desistir ou não se esforçar para melhorar os processos de previsão para operações, por não se conseguir acertar essas previsões. Quando não se necessita de previsões perfeitas, mas previsões consistentemente melhores do que as da concorrência.

As condições de negócios resultantes do atual ambiente competitivo, das rápidas mudanças tecnológicas e de preocupações ambientais crescentes exercem pressão sobre a capacidade de as empresas gerarem previsões precisas. As previsões raramente são perfeitas, entretanto, a utilização de métodos de previsão podem torná-las melhores (KRAJEWSKI; RITZMAN; MALHOTRA, 2009).

A seleção de uma abordagem de previsão (qualitativa, causal ou baseado em séries temporais), segundo Fernandes e Godinho Filho (2010) passa por quatro importantes pontos:

1. **A existência ou não de dados**: caso não existam dados deve-se usar uma abordagem qualitativa para a previsão de demanda.
2. **A possibilidade de coleta desses dados**: caso sejam muito difíceis para serem coletados, deve-se usar uma abordagem qualitativa.
3. **A natureza dos dados (qualitativos ou quantitativos)**: se forem somente qualitativos, então também a abordagem qualitativa será a mais adequada. Caso existam dados quantitativos, deve-se verificar a existência de algum fator causal.
4. **A existência ou não de fatores causais**: fator causal é algum fator que influencia os dados de uma forma conhecida facilitando a previsão. Se existir algum fator a abordagem causal é a mais adequada, caso existam dados quantitativos e não exista um fator causal, então a abordagem baseada em séries temporais deve ser utilizada.

Cada uma das três abordagens é composta de diversos métodos de previsão. Então, uma vez identificada à melhor abordagem, deve-se escolher o método de previsão. O Quadro

4 apresenta alguns dos métodos ou técnicas de previsão mais comuns, organizados de acordo com as abordagens de previsão, incluindo uma quarta abordagem: os modelos de simulação.

Quadro 4 - Técnicas de previsão e modelos comuns

I. Qualitativo	Subjetiva; de julgamento. Baseada nas estimativas e opiniões.
Senso Comum	Obtém uma previsão mediante a compilação de opiniões daqueles no final da hierarquia que lida com o que está sendo previsto. Por exemplo, uma previsão geral de vendas poderá ser obtida combinando-se as opiniões de cada vendedor, que seja mais próxima de seu território.
Pesquisa de mercado	Propõe-se colher dados de várias maneiras (pesquisas, entrevistas etc.) para testar as hipóteses sobre o mercado. É usada geralmente para prever as vendas de longo alcance e de produtos novos.
Consenso do painel	Troca aberta livre nas reuniões. A ideia é que as discussões em grupo produzirão previsões melhores do que um indivíduo. Os participantes podem ser executivos, vendedores ou clientes.
Analogia histórica	Liga o que está sendo previsto a um item similar. É importante no planejamento de produtos novos, onde uma previsão poderá ser obtida pelo uso da história de um produto similar.
Método Delphi	Grupos de especialistas respondem ao questionário. Um moderador compila os resultados e formula um novo questionário que é submetido para o grupo. Assim, há um processo de aprendizado para o grupo, conforme esse recebe novas informações e não há influência da pressão de grupos ou de indivíduos dominantes.
II. Análise de séries temporais	Baseada na ideia de que a história de ocorrências com o tempo pode ser usada para prever o futuro.
Média móvel Simples	Pondera-se um período de tempo contendo um número de pontos dividindo a soma dos valores dos pontos pelo número de pontos. Cada um deles, portanto, tem influência igual.
Média móvel ponderada	Pontos específicos podem ser ponderados mais ou menos do que outros, como parecem adequados pela experiência.
Média ponderada exponencial	Pontos recentes são ponderados mais com o peso, declinando exponencialmente à medida que os dados se tornam mais antigos.
Análise de regressão	Adapta uma linha reta aos dados passados, geralmente relacionando os valores de dados no tempo. A técnica de adaptação mais comum é a de mínimos quadrados.

Continuação do Quadro 4 - Técnicas de previsão e modelos comuns.

Técnica de *Box Jenkins*	Muito complicada, mas aparentemente a técnica estatística mais precisa disponível. Relaciona uma classe de modelos estatísticos aos dados e adapta o modelo à série temporal usando as distribuições posteriores de Bayesian.
Série temporal *Shiskin*	Um método eficaz para decompor uma série temporal em sazonais, tendências e irregulares. Precisa de no mínimo três anos de história. Muito boa na identificação de pontos críticos, por exemplo, nas vendas das empresas.
Projeções de tendência	Adapta uma linha de tendência matemática aos pontos e a projeta no futuro.
III. Causal	Tenta entender o sistema básico e ao redor do item que está sendo previsto. Por exemplo, as vendas poderão ser afetadas pela propaganda, qualidade e competidores.
Análise de regressão	Similar ao método dos mínimos quadrados na série temporal, mas poderá conter variáveis múltiplas. A base é que a previsão é causada pela ocorrência de outros eventos.
Modelos de econometria	Tentativas para descrever alguns setores da economia mediante uma série de equações independentes.
Modelos de entrada/saída	Foco nas vendas de cada indústria para as outras empresas e governo.Indica as mudanças nas vendas que uma indústria produtora poderá esperar exceto por causa das mudanças nas compras por outra indústria.
Indicadores	Estatísticas que se movimentam na mesma direção que a série que está sendo prevista, mas se movimentam antes da série, como um aumento de preço da gasolina indica uma queda futura na venda de carros grandes.
IV. Modelos de Simulação	Modelos dinâmicos, geralmente computacionais, que permitem que a pessoa que está fazendo a previsão faça suposições sobre as variáveis internas e o ambiente externo no modelo. Dependendo das variáveis no modelo, a pessoa faria perguntas como o que aconteceria com a minha previsão se o preço aumentasse em 10%? Qual efeito uma leve recessão nacional teria na minha previsão?

Fonte: Chase, Jacobs e Aquilano (2006, p.454)

Quando se faz uma previsão, uma boa estratégia é utilizar-se de dois ou três métodos (de acordo com a abordagem) e procurar nos resultados obtidos um ponto de vista de bom senso. Como as mudanças esperadas na economia afetarão a previsão? Existem mudanças no comportamento do cliente? A revisão contínua e a atualização, em vista de novos dados, são básicas para previsões de sucesso (CHASE; JACOBS; AQUILANO, 2006).

É necessária muita coordenação para gerir as demandas, sejam elas dependentes ou independentes. O propósito dessa gestão consiste em coordenar e controlar todas as origens de demanda para que o sistema produtivo possa ser usado eficientemente e o produto possa ser entregue no prazo. O departamento de operações precisa das previsões para planejar os níveis de produto, compra de serviços e materiais, mão de obra e cronogramas de produção. A previsão de demanda total geralmente provém do marketing, sendo essa área uma fonte fundamental para informações de previsão de vendas porque está mais próxima dos clientes (CHASE; JACOBS; AQUILANO, 2006; KRAJEWSKI; RITZMAN; MALHOTRA, 2009).

Definindo, de acordo com Slack, Chambers e Johnston (2009) demanda dependente é aquela relativamente previsível devido a sua dependência de alguns fatores conhecidos, ela é originada a partir da demanda de outros produtos ou serviços, esse tipo de demanda não precisa de previsão, necessitará apenas de uma tabulação. Já a demanda independente é aquela da qual não se tem qualquer visibilidade firme antecipada dos pedidos dos clientes.

Em relação à demanda independente uma empresa pode, segundo Chase, Jacobs e Aquilano (2006), assumir um papel ativo para influenciar a demanda ou assumir um papel passivo e simplesmente atendê-la. Como as variações de demanda são reais, o planejamento deverá incluir flexibilidade suficiente para lidar com essas variações (fontes alternativas de suprimento, trabalhadores que possam realizar uma variedade de tarefas diferentes, planejamentos frequentes durante a alta demanda). O plano agregado de produção ou planejamento agregado determina os recursos necessários (mão de obra, estoque etc.) para a produção atender a demanda de seus produtos ou serviços, ou seja, possibilita buscar formas de se influenciar a demanda de maneira a torná-la mais constante ao longo dos períodos (FERNANDES; GODINHO FILHO, 2010; TUBINO, 2008).

Existem fatores internos e externos que compõem o ambiente de planejamento da produção, os fatores que fazem parte do ambiente externo, normalmente estão fora do controle direto do planejador da produção, mas em algumas empresas a demanda pelo produto poderá ser administrada, por meio da estreita colaboração entre as áreas de marketing e produção. Os fatores internos diferem no seu controle, a capacidade física é quase sempre fixada no curto prazo e não pode ser continuamente aumentada; pode haver restrições financeiras, entre outros.

A Figura 10 ilustra alguns fatores internos e externos que compõem o ambiente de planejamento da produção.

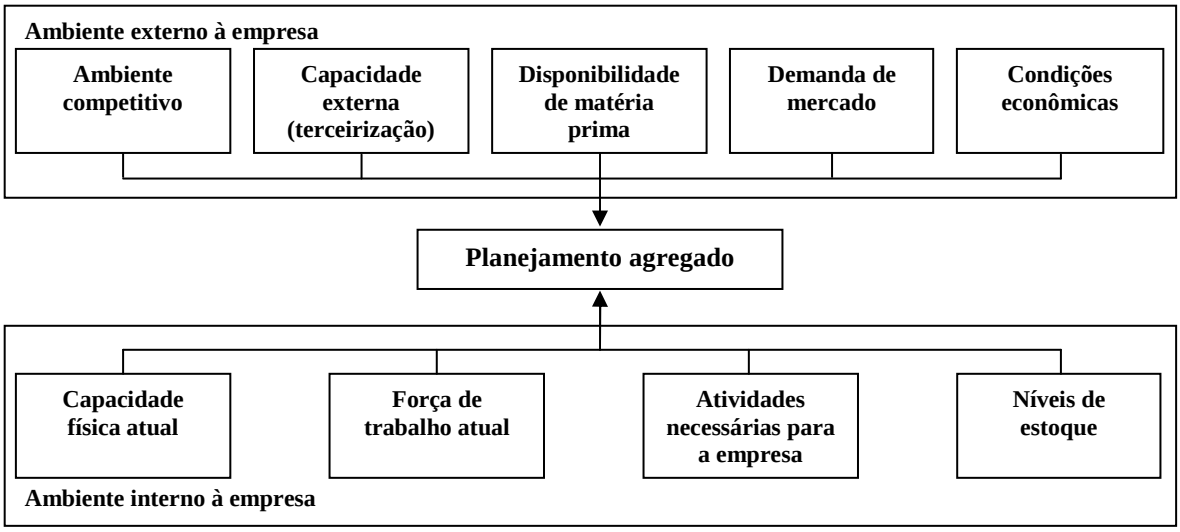

Figura 10 - Fatores que compõem o ambiente externo e interno do planejamento agregado
Fonte: adaptado de Russell e Taylor III (2006)

Há limites sobre quanta demanda pode ser controlada, os planejadores da produção podem programar uma estratégia ou uma combinação delas de maneira a administrar esses fatores.

Segundo Chase, Jacobs e Aquilano (2006) existem essencialmente três estratégias:

- **Acompanhamento da demanda**: combina a taxa de produção com a taxa de pedidos, contratando e demitindo funcionários à medida que a taxa de pedidos varia. O sucesso depende do fato de haver um grupo de candidatos, facilmente treinados, com o qual se pode contar de acordo com o aumento do volume.
- **Mão de obra estável (horas de trabalho variáveis)**: modificar a produção através da variação do número de horas trabalhadas por meio de horários de trabalho flexíveis ou horas extras. Dessa forma, se consegue combinar as quantidades de produção aos pedidos.
- **Capacidade constante**: manter uma mão de obra estável trabalhando numa taxa constante de produção. As faltas e excessos são absorvidos pelos níveis de estoque flutuantes, pelos pedidos em atraso e vendas perdidas.

O mais usual nas indústrias é a utilização de uma estratégia mista, ou seja, a combinação de duas ou mais estratégias usadas em combinação, afinal cada uma das estratégias tem pontos positivos e negativos. Na primeira existem impactos emocionais relacionados à política de demissão, quando o nível de pedidos em atraso baixa, é natural que os funcionários diminuam o ritmo de produção de forma a mantê-los e assim evitarem o desligamento. A segunda estratégia proporciona continuidade para a mão de obra e evita

desgastes emocionais e custos relacionados a demissão e contratação e treinamento de novos colaboradores. Na terceira estratégia os funcionários se beneficiam das horas estáveis e a empresa dos custos relacionados a esse fato, já os produtos estocados podem ser tornar obsoletos e os custos de atendimento ao cliente e estoque aumentam.

A terceirização é uma alternativa à estratégia de acompanhamento de demanda, assim as contratações e demissões são traduzidas em terceirizar ou não. Nesse caso, o relacionamento com o fornecedor deve ser forte de maneira a se manter o controle sobre a programação e qualidade dos produtos ou serviços prestados.

A Figura 11 ilustra as estratégias da produção comentadas acima.

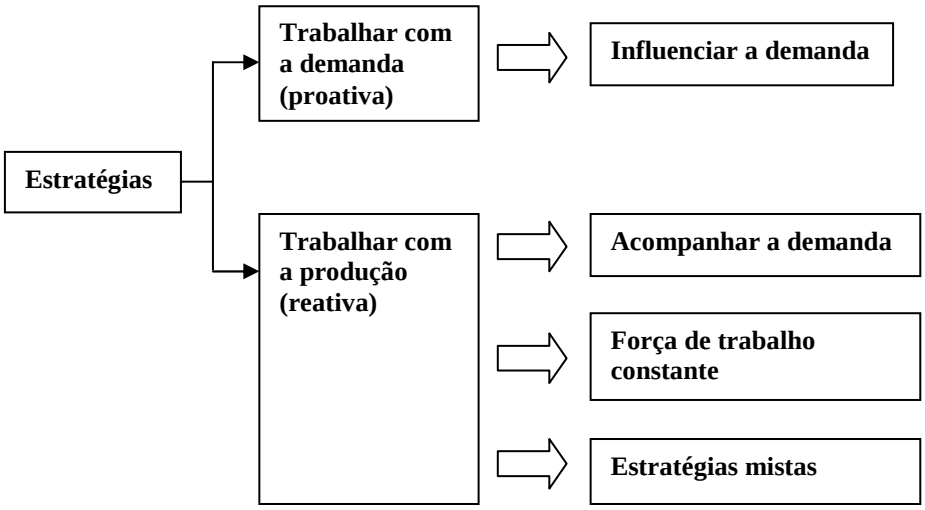

Figura 11 - Estratégias de planejamento agregado
Fonte: adaptado de Fernandes e Godinho Filho (2010)

Segundo Fernandes e Godinho Filho (2010) uma classificação importante a respeito dos sistemas produtivos é a classificação baseada na estratégia de resposta à demanda. Esses autores apresentam seis estratégias, a saber:

- *Make to stock* (MTS → produção para estoques com base em previsão de demanda);
- *Quick response to stock* (QRTS → produção para estoque com base em uma rápida reposição);
- *Assembly to order* (ATO → montagem sob encomenda);
- *Make to order* (MTO → fabricação sob encomenda, com estoque de insumos);
- *Resources to order* (RTO → recursos e insumos sob encomenda); e
- *Engineering to order* (ETO → projeto sob encomenda).

A Figura 12 apresenta como se comportam alguns importantes fatores frente à escolha da estratégia de resposta à demanda.

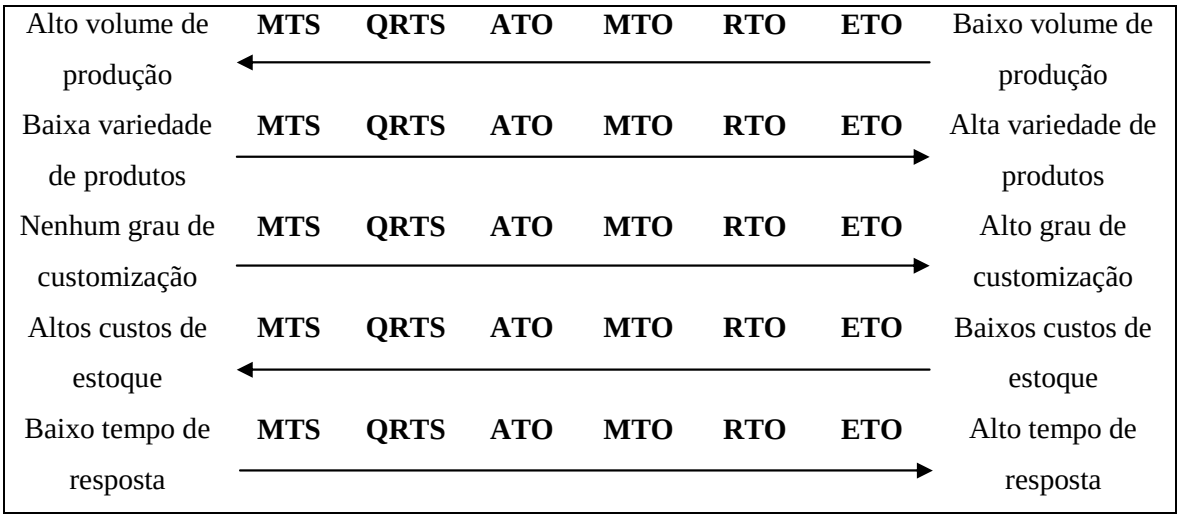

Figura 12 - Comportamento de alguns fatores x estratégia de atendimento a demanda
Fonte: adaptado de Fernandes e Godinho Filho (2010)

No nível tático, o planejamento agregado da produção servirá de base para desenvolver o planejamento-mestre da produção, onde as informações serão desmembradas de forma a permitir a programação do sistema produtivo (TUBINO, 2008).

Segundo Slack, Chambers e Johnston (2009) a fase mais importante do planejamento e controle de uma empresa é o programa-mestre de produção *Master Production Schedule* (MPS). Ele contém uma declaração da quantidade e momento em que os produtos finais devem ser produzidos, direcionando toda operação em termos do que deve ser montado, manufaturado e comprado.

Um fator muito importante na geração do MPS é a estratégia de resposta a demanda adotada, a saber: produção para estoque, produção para estoque com base em uma rápida reposição, montagem sob encomenda, fabricação sob encomenda, com estoque de insumos, recursos e insumos sob encomenda e projeto sob encomenda.

No caso da reposição rápida do estoque (QRTS) não é elaborado um MPS, nesse caso definem-se os níveis de estoques de produtos finais e a política de reabastecimento dos mesmos.

Já nos casos fabricação sob encomenda com estoque de insumos, recursos e insumos sob encomenda, e projeto sob encomenda (respectivamente MTO, RTO e ETO), a produção individual de cada item é baseada em pedidos de clientes. Assim, o MPS para esses itens é desenvolvido com base na carteira de pedidos.

Nos casos de produção para estoque com base em previsão de demanda e montagem sob encomenda (respectivamente MTS e ATO), o MPS é feito com base na desagregação do planejamento agregado ou na previsão de demanda de curto prazo. A Figura 13 apresenta os dados de entrada do MPS em função da estratégia de resposta à demanda.

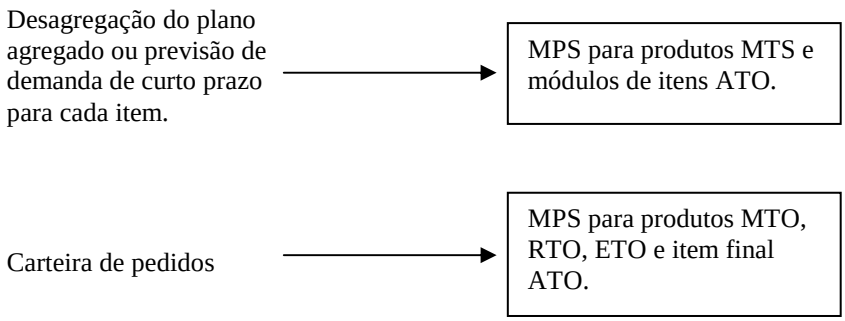

Figura 13 - Dados de entrada para a geração do MPS x estratégia de resposta à demanda
Fonte: adaptado de Fernandes e Godinho Filho (2010)

Os dois principais objetivos da desagregação do planejamento agregado são servir de base para o MPS em produtos MTS e ATO, e controlar o suprimento de itens com *lead time* de suprimento longo (itens de *lead time* de suprimento longo não podem esperar a chegada dos pedidos de clientes para serem comprados).

Em termos de prazos, o MPS realiza duas funções básicas dentro do planejamento e controle da produção: uma é direcionar a programação para atender aos pedidos dos clientes no curto prazo, e a outra é permitir a análise e validação da capacidade do sistema produtivo em atender à demanda futura.

Quanto maior o nível de repetição do sistema de produção, mais fácil o estabelecimento de um MPS. Quanto mais instável o sistema de produção e seu ambiente, mais se torna necessário trabalhar com horizonte de curta duração. A determinação dos intervalos de tempo que irão compor o MPS está associada à velocidade de fabricação dos itens incluídos no plano-mestre de produção (TUBINO, 2008; FERNANDES; GODINHO FILHO, 2010).

Resumindo, as quatro principais atividades do controle da produção são:

1. Programar a produção em termos de itens finais, elaborando o programa-mestre da produção;
2. Organizar e explodir as necessidades em termos de componentes e materiais;
3. Controlar a emissão e liberação de ordens de produção e compra, determinando se e quando liberar as ordens;
4. Programar e sequenciar as tarefas nas máquinas.

São os sistemas de coordenação de ordens de produção e compras (SCO) que programam e explodem as necessidades em termos de componentes e materiais; controlam a emissão e liberação das ordens de produção e compra; programam e sequenciam as tarefas nas máquinas. Assim, um SCO basicamente coordena as ordens de produção e compras no chão de fábrica (FERNANDES; GODINHO FILHO, 2010).

2.1.2 Sistemas de coordenação de ordens de produção e compra (SCO)

Por meio da classificação dos sistemas produtivos pode-se selecionar o SCO mais adequado, podendo classificá-los a partir desse conceito.

Quando um SCO toma as decisões de o que, quando, quanto comprar, produzir e entregar; e onde e como produzir com base no estoque, se diz que o SCO é um sistema controlado pelo nível de estoque (CNE), o qual puxa a produção.

Quando é impossível manter estoques de produtos finais para atender os clientes, então se diz que o SCO nesse caso é um sistema de pedido controlado (sistemas de programação por contrato, sistemas de alocação de carga por encomenda).

Quando um SCO converte as necessidades dadas em produtos finais (especificadas no MPS) para necessidade em termos de itens componentes ou materiais comprados, então se diz que esse sistema é de fluxo programado (MRP, OPT).

Quando a informação caminha em uma direção e o fluxo de materiais caminha em outra, nesse caso o SCO puxa a produção. Se o fluxo de informações e materiais caminharem na mesma direção, então o SCO empurra a produção (*kanban*).

Há várias possibilidades de coordenar a produção, umas mais adequadas a certas situações e outras mais adequadas a outras características de unidades produtivas ou de itens.

O Quadro 5 os SCO mais adequados de acordo com os níveis de repetição dos sistemas de produção.

Quadro 5 - Níveis de planejamento e programação da produção

Nível	Item é considerado	Período considerado	Horizonte usual de planejamento/programação
Planejamento agregado da produção	Família de produtos finais	Mês	12 meses (vários meses)
Planejamento desagregado da produção	Produto final	Mês	12 meses (vários meses)
Programa-mestre da produção	Produto final ou módulo	Semana	Várias semanas
Programação no nível de componentes e materiais	Módulo ou componente ou matéria prima	Semana ou dia	Algumas ou várias semanas ou dias
Programação de operações	Operação	Dia ou turno ou hora	Alguns dias ou várias horas

Fonte: adaptado de Fernandes e Godinho Filho (2010)

Os três principais SCO identificados na literatura do PCP: *Material Requirements*

Planning (MRP), *Just-in-time* (JIT), *kanban* e o *Optimized Production Technology* (OPT).Segundo Hemmondharop (2001), essas filosofias de gerenciamento têm sido as técnicas mais difundidas e utilizadas em uma variedade de indústrias durante as últimas décadas.

2.2 Controlede chão de fabrica: programação e*scheduling*

Segundo Fernandes e Godinho Filho (2010), o controle do chão de fábrica *shop floor control* (SFC) compreende três atividades principais:

- Liberação;
- Programação de operações (*scheduling*); e
- Apontamento da produção.

A primeira atividade envolve a liberação das ordens de compra, fabricação, montagem e depende do sistema de coordenação de ordens (SCO) utilizado. Como descrito anteriormente existem várias possibilidades de coordenar a produção e seria simplista imaginar que existam somente o MRP, OPT, JIT e *kanban*. Retomando, um SCO pode ser selecionado de acordo com as características da unidade produtiva, ou partindo das características dos itens ou combinando essas duas abordagens.

O apontamento da produção é composto pelo monitoramento da realização das ordens de forma a garantir que o programa de produção seja cumprido ou desvios identificados e corrigidos; pelo cálculo de indicadores de desempenho, ou seja, medidas que permitam o progresso das ordens, avaliar o desempenho do processo de produção, mão de obra ou de qualquer outro objetivo; e realimentação (*feedback*) das informações para os responsáveis pelo controle da produção sobre o que está ocorrendo no chão de fábrica e sobre os valores dos indicadores. Este é o último conjunto de atividades do SFC.

A programação é a segunda atividade a ser realizada e ocorre quando se deve alocar e seqüenciar **n** tarefas em **m** recursos, enquanto que o sequenciamento de tarefas ocorre quando se precisa priorizar a ordem da próxima (tarefa ou *job*) a ser executada em um recurso. Portanto, o problema de sequenciamento de tarefas é um caso particular da programação de operações e será discutido detalhadamente deste ponto em diante.

2.2.1 *Scheduling*

Segundo Herrmann (2006) *scheduling* começou de forma simples. No início, listavam-se apenas quando o trabalho ou ordem deveria começar e quando deveria terminar. As

programações não forneciam quaisquer informações sobre quanto tempo o trabalho levaria para ser realizado ou sobre o tempo exigido para as operações individuais. Frederick Taylor separou o planejamento da execução justificando assim, o uso de métodos de *scheduling*, que se tornaram críticos devido ao crescimento e complexidade das organizações industriais.

Mas, esta história começa nos primeiros gráficos desenvolvidos por Henry Gantt, que em seu trabalho publicado em 1916, discute explicitamente *scheduling*, especialmente no ambiente *job shop*. Ele propõe entregar ao capataz todo dia uma ordem de trabalho que terá uma lista ordenada dos trabalhos que deverão ser feitos naquele dia, Gantt projetou seus gráficos de forma que o capataz ou outros supervisores, pudessem rapidamente saber se a produção estava de acordo com a programação, à frente dela, ou atrasada. Os *softwares* de gerenciamento de projeto modernos incluem atualmente esta função (GANTT, 1916 *apud* HERRMANN, 2006).

Gantt (1919 *apud* HERRMANN, 2006) apresentou dois princípios para seus gráficos:

1) Mensurar as atividades por meio da quantidade de horas necessárias para completá-la;
2) Os espaços no gráfico podem ser usados para representar a quantidade de atividades que poderiam ter sido feitas naquele tempo.

Pode-se dizer que Gantt foi um pioneiro em desenvolver gráficos para visualizar a programação da produção. Ele inovou usando tempo (não quantidade justa) como uma forma para medir tarefas, e barras horizontais para representar o número de partes produzidas e registrar tempo de trabalho. A Figura 14 apresenta um gráfico de Gantt, o problema considera a produção de dois itens A e B em um *flow shop* (sem *setup*) sobre um horizonte discreto. A rotina consiste em três máquinas M1, M2, e M3. O tempo de processamento por unidade é: $p^u_{A1}= 2$; $p^u_{A2}= 1$; $p^u_{A3}= 1$; $p^u_{B1}= 3$; $p^u_{B2}= 1$; $p^u_{B3}= 2$.

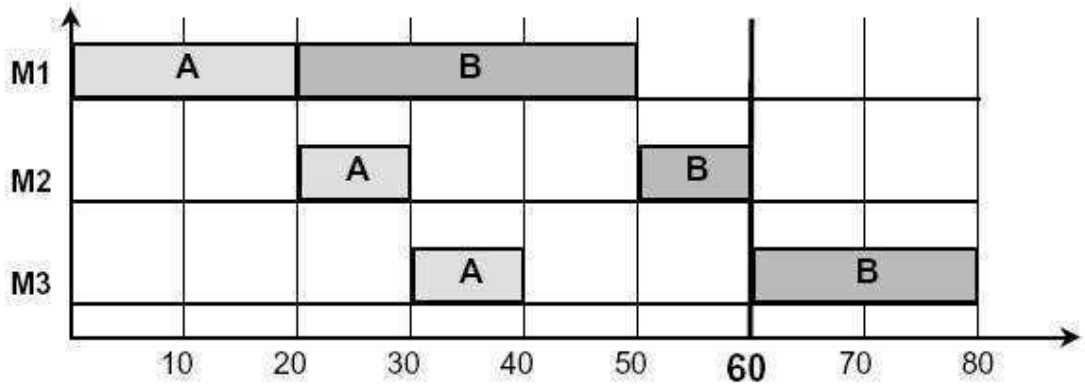

Figura 14 - Gráfico de Gantt
Fonte: adaptado de Dauzère-pérès e Lasserre (2002)

De acordo com Leung (2004) *scheduling* preocupa-se com a distribuição de recursos escassos para atividades com o objetivo de otimizar uma ou mais medidas de desempenho. Segundo Pinedo (2008) *scheduling* é um processo de tomada de decisão que é usado de maneira regular em muitas organizações, seu negócio é a alocação de recursos para tarefas de acordo com os períodos de tempo determinados e sua meta é otimizar um ou mais indicadores.

Esses autores indicam que dependendo da situação, recursos e atividades podem apresentar muitas formas diferentes. Recursos podem ser identificados como máquinas disponíveis em fábricas, computadores, dispositivos de entrada e saída em sistemas de computador, pistas em um aeroporto, mecânica em uma loja de conserto de automóvel, entre outras. As atividades podem ser várias operações no processo industrial, a execução de um programa de computador, aterrissagens e partidas em um aeroporto, consertos de carro em uma loja de conserto de automóvel, e assim por diante. E também afirmam que existem muitas medidas de desempenho para otimizar, por exemplo, um objetivo poderá ser minimizar o *makespan*, enquanto outro pode ser minimizar o número de tarefas atrasadas.

A teoria de *scheduling* recebeu muita atenção de vários pesquisados de várias áreas (ciências sociais aplicadas, engenheiros, matemáticos) desde o início da década de 1950 e vários livros e artigos foram publicados sobre o assunto desde então, por exemplo, Conway, Maxwell e Miller (2003); Baker (1974); Rinnooy Kan (1976); French (1982). Os problemas estudados nesse período eram relativamente simples e vários algoritmos eficientes foram desenvolvidos com a intenção de oferecer soluções ótimas, os trabalhos mais notáveis do período em questão são os realizados pelos pesquisadores Jackson (1955), Johnson (1954) e Smith (1956) (MACCARTHY; LIU, 1993; LEUNG, 2004).

Com o passar do tempo, os problemas encontrados se tornaram mais sofisticados, e a resolução desses problemas mais difíceis exigiu uma abordagem diferente. A técnica *branch-and-bound* surgiu em meados dos anos de 1960, algoritmos nesse momento implicitamente enumeravam todas as soluções possíveis e achavam uma solução ótima. Ao mesmo tempo, técnicas de relaxamento, de geração de coluna para programação linear, e programação de restrição foram desenvolvidas para resolver problemas de programação de linear inteira (HERRMANN, 2006; RAGSDALE, 2009).

Com o advento da teoria da complexidade, na década de 1970, os pesquisadores começaram a perceber que muitos problemas poderiam ser inerentemente difíceis de resolver. Nesse período, vários problemas de *scheduling* eram apresentados como sendo do tipo *NP-hard* (NP-difícil, ou NP-complexo), e por um bom tempo os algoritmos que poderiam achar

soluções ótimas para estes problemas, não puderam existir. Os avanços em TI tornaram possíveis os sistemas de *scheduling* baseado em computador; algoritmos inovadores foram desenvolvidos a partir de então. Segundo Leung (2004) bons algoritmos de *scheduling* permitem a redução de custos do processo produtivo das empresas, permitindo que as mesmas se mantenham competitivas.

Nas décadas de 1980 e 1990, devido à necessidade dos tomadores de decisão de agilizar a obtenção de solução, os algoritmos de procura que podiam achar resultados ótimos se tornaram importantes. Foram desenvolvidos então algoritmos de busca local como *simulated annealing*, e *tabu search.* Outras inovações incluíram algoritmos genéticos, modelos de programação matemática e inteligência artificial. Muitos anos de pesquisa em métodos de otimização criou um conjunto grande de algoritmos poderosos que podem ser aplicados para gerar *schedules* (HERRMANN, 2006; LEUNG, 2004).

A TI gerou um grande impacto em como o sequenciamento da produção é feito. Entre os muitos benefícios da TI está a habilidade de executar algoritmos complexos, que permitem combinar regras de *scheduling* com a programação de produção de capacidade finita e guiar as decisões que devem ser tomadas em tempo real. O desenvolvimento de algoritmos melhores é deste modo uma parte importante da história do sequenciamento e programação da produção.

2.2.2 *Scheduling*: estrutura e notação

Pode-se dizer que os dois problemas chave no *scheduling* são: prioridades e capacidade. Em outras palavras, o que deve ser feito primeiro? E quem deverá fazer isto? Os problemas de *scheduling* consideram o número de trabalhos e o número de máquinas como sendo finitos, como deve satisfazer um grande número de restrições de tempo e de relações entre as atividades e recursos, caso os recursos sejam ilimitados, um problema de *scheduling* não existirá.

Segundo Conway, Maxwell e Miller (2003) existe uma notação com quatro parâmetros que deve ser usada para identificar problemas de *scheduling*: A/B/C/D.

- **A**: apresenta o número de tarefas e respectivas operações que serão realizadas. Para problemas dinâmicos, ele identificará a probabilidade dos tempos entre chegadas. Para problemas estáticos, especificará o número de trabalhos, assumindo chegadas simultâneas.
- **B**: descreve o número de máquinas disponíveis.
- **C**: descreve o tipo de ambiente (fluxo padrão) em que as operações serão realizadas.

- **D**: descreve o critério pelo qual o sequenciamento será avaliado.

A notação que será descrita no Quadro 6 é a encontrada nos livros de *scheduling* de Pinedo (2008); Leung (2004); Conway, Maxwell e Miller (2003), bem como Baker (1974).

Quadro 6 - Principais notações de *scheduling*

d_i	Prazo de entrega (*due date*) da tarefa **i**
r_i	Data de liberação da tarefa **i**
N	Número de *jobs*
M	Número de operações da tarefa **i**
Se m = 1→ p_i	Tempo de processamento da tarefa **i**
Se m ≥ 2 → p_{ij}	Tempo de processamento da j-ésima operação da tarefa **i**
W_{ij}	Tempo de espera da tarefa **i** no momento que terminou a operação (j -1) até ser iniciada a operação **j**
W_i	Tempo total de espera da tarefa **i** → $W_i = \sum_{j=1}^{m} W_{ij}$
C_i	Tempo de conclusão da tarefa **i** → $C_i = r_i + \sum_{j=1}^{m} P_{ij} + \sum_{j=1}^{m} W_{ij}$
C	Tempo total de conclusão das **n** tarefas → $C = \sum_{i=1}^{n} C_i$
F_i	Tempo de permanência (ou de fluxo) da tarefa **i** → $F_i = C_i - r_i$; Note que se $r_i = 0$,então $F_i = C_i$
F	Tempo total de fluxo das **n** tarefas → $F = \sum_{i=1}^{n} F_i$
$\bar{F}$	Tempo médio de fluxo → $\bar{F} = F/n$
L_i	Atraso da tarefa **i** → $L_i = C_i - d_i$
L	Atraso total das **n** tarefas = $\sum_{i=1}^{n} L_i$
$L_i < 0$	Significa que a tarefa **i** ficou pronta antes do prazo
T_i	Tempo de atraso da tarefa **i** → $T_i = \max\{0, L_i\}$
T	Tempo total de atraso das **n** tarefas → $T = \sum_{i=1}^{n} T_i$

Fonte: adaptado de Pinedo (2008); Leung (2004); Conway, Maxwell e Miller (2003); Baker (1974)

Pode-se observar nas notações apresentadas que as letras maiúsculas são variáveis de resposta e as minúsculas são dados (parâmetros) para um determinado problema.

Resumidamente, os problemas de programação podem ser descritos da seguinte maneira, conforme relatam Maccarthy e Liu (1993).

- **n** tarefas $\{j_1, ..., j_n\}$ devem ser processadas por **m** máquinas$\{M_1, ..., M_m\}$.
- Cada tarefa deve ser processada em cada máquina somente uma vez.
- O processamento de uma tarefa **j** em uma máquina **m** é chamado de operação com duração p_{jm} conhecida
- Para cada operação O_{ij} existe um tempo T_{ij} de processamento associado.
- O instante em que a tarefa **j** está disponível para processamento é conhecido.

Nesse contexto programar a produção consiste em definir a ordem de entrada das tarefas a serem executadas em cada máquina. Assim, a programação da produção tem como um de seus objetivos otimizar as medidas de desempenho. As suposições seguintes aparecem frequentemente na literatura (BAKER, 1974).

1. Máquinas estão sempre disponíveis e nunca quebram.
2. Nenhuma máquina pode processar mais de uma operação por vez.
3. Tempos de processamento e restrições tecnológicas são conhecidas.
4. Tempos de *setup* são independentes da programação e estão incluídos nos tempos de processamento.
5. Cada operação, uma vez iniciada, deve ser completada antes que outra operação seja iniciada naquela máquina.
6. Duas operações de uma mesma tarefa não podem ser executadas simultaneamente.
7. Cada tarefa deve ser processada até seu término.

As restrições tecnológicas são determinadas, principalmente, pelo padrão de fluxo dos trabalhos nas máquinas. Os fluxos são identificados conforme relacionado:

***Job Shop* (G)**: em um *job shop* com **m** máquinas, cada trabalho tem sua própria rota para seguir (existe uma ordem para que as peças passem pelas máquinas). Pode visitar algumas máquinas mais de uma vez e pode não visitar outras máquinas, o roteiro de fabricação não é necessariamente o mesmo para todas as peças. Este problema é matematicamente complexo e de difícil tratamento (*NP-hard*).

***Flow Shop* (F)**: em um *flow shop* com **m** máquinas, as máquinas são linearmente ordenadas e todas as peças têm o mesmo roteiro de fabricação e passam nas mesmas máquinas. Tem-se um roteiro de fabricação único onde as **n** peças utilizam **m** máquinas na mesma ordem (da primeira máquina até a última máquina).

***Open Shop* (O)**: em um *open shop* com **m** máquinas, neste caso não há uma ordem na fabricação. É um modelo menos restrito que *o job shop* e o *flow shop*. A ordem de execução das tarefas componentes de um determinado trabalho é livre.

***Flow shop* permutacional (Fperm):** em um *flow shop* com **m** máquinas, em que a ordem de processamento dos trabalhos é limitada para ser a mesma em todas as máquinas.

As possibilidades de máquinas no sistema produtivo são as que seguem:

Maquina única (1): existe uma única máquina disponível no sistema

Máquinas paralelas (k-paralela): existem **m** máquinas paralelas disponíveis, geralmente idênticas, cada trabalho exige uma operação única e pode ser processado em quaisquer uma das **m** máquinas.

Quando estágios de processamento são considerados em lugar de máquinas, as definições seguintes são úteis:

***Job shop* com máquinas múltiplas (G,k-paralela):** um *job shop* com **m** máquinas paralelas idênticas em cada estágio, e qualquer tarefa precisa ser processada em somente uma destas máquinas.

***Flow shop* com máquinas múltiplas (F,k-paralela):** um *flow shop* com **m** máquinas paralelas idênticas em cada estágio.

A Figura 15 ilustra a relação entre os diferentes ambientes e quantidades de máquinas.

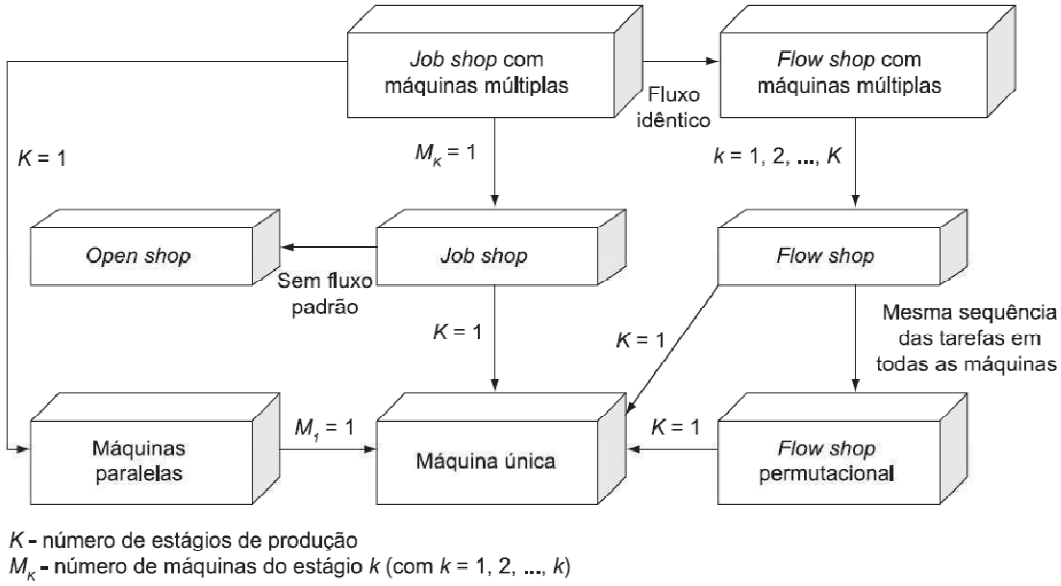

Figura 15 - Relação entre ambientes e máquinas
Fonte: adaptado de Maccarthy e Liu (1993)

As abreviações apresentadas no Quadro 7 normalmente são usadas para representar restrições adicionais que podem ocorrer em ambientes de *scheduling* mais complexos.

Quadro 7 - Abreviações

r_j	Trabalhos com diferentes tempos de finalização
Str	*String* (sequência) de trabalhos
prec	Restrições de precedência
Unit	Tempos de processo da unidade
Eq	Tempo de processo igual para todos os trabalhos
depend	Trabalhos dependentes
setup	sequência com *setup* dependente

Fonte: adaptado de Maccarthy e Liu (1993)

A dificuldade teórica dos problemas de programação aumenta à medida que são consideradas mais máquinas, e não à medida que são processadas mais tarefas; assim, a única restrição sobre **n** é que este é um número conhecido e finito. A próxima etapa de complexidade ocorre quando duas ou mais tarefas precisam ser processadas em duas máquinas em uma sequência comum. A programação mais complexa é caracterizada pelos múltiplos centros de máquinas, processando uma variedade de tarefas diferentes que chegam intermitentemente durante todo o dia. Se houverem **n** tarefas para serem processadas em **m** máquinas, e todas as tarefas são processadas em todas as máquinas, então existem $(\mathbf{n}!)^{\mathbf{m}}$ programas alternativos para este conjunto de tarefas (CHASE; JACOBS; AQUILANO, 2006).

Segundo Pinedo (2008) durante várias décadas, foram realizadas, pesquisas sobre regras de prioridade (*dispatching*) e muitas regras diferentes foram estudadas na literatura. Essas regras podem ser classificadas de diversas maneiras, como por exemplo, uma distinção pode ser feita entre regras estáticas e dinâmicas; outra forma de classificá-las relaciona-se com as informações na quais elas se baseiam (regra local ou global).

Regras de prioridades são úteis quando se deseja encontrar um bom sequenciamento em relação a um único objetivo, como o *makespan*, o tempo de conclusão total ou o atraso máximo. No entanto, os objetivos no mundo real são muitas vezes mais complicados. Por exemplo, um objetivo de um problema real pode ser uma combinação de vários objetivos básicos e também em função do tempo ou em função do conjunto de trabalhos em espera para processamento. O Quadro 8 apresenta uma visão geral de algumas das regras mais conhecidas. Todas essas regras possuem variações que podem ser aplicados em cenários mais complexos.

Quadro 8 - Regras de prioridades

SPT (*Shortest Processing Time*)	*jobs* são sequenciados na ordem crescente de seus tempos de processamento.
LPT (*Longest Processing Time*)	*jobs* são sequenciados na ordem decrescente de seus tempos de processamento.
EDD (*Earliest Due Date*)	seleciona-se para processamento a tarefa com a data de entrega mais cedo. Esta regra tende a minimizar o atraso máximo entre as entidades da fila de espera.
FCFS (*First Come, First Served*)	tarefas são processadas na mesma sequência em que chegam na instalação.
MWKR (*Most Work Remaining*)	seleciona-se a operação correspondente a tarefa que tem a maior quantidade de trabalho remanescente a ser executado.
LWKR (*Least Work Remaining)*	seleciona-se a operação correspondente a tarefa que tem a menor quantidade de trabalho remanescente a ser executado.
MOPNR (*Most Operations Remainning)*	seleciona-se a operação que tem o maior número de operações sucessoras.
RANDOM	Seleciona-se a operação randomicamente, ou seja, de maneira aleatória.

Fonte: adaptado de Pinedo (2008); Conway, Maxwell e Miller (2003)

Segundo Maccarthy e Liu (1993) vários métodos foram desenvolvidos para resolver problemas de *scheduling*. Eles são basicamente de dois tipos: os métodos de solução ótima e os métodos heurísticos.

Maccarthy e Liu (1993) subdividem os métodos de solução ótima em dois tipos: os métodos ótimos enumerativos e os métodos ótimos eficientes.

De acordo com Pacheco e Santoro (1999) dentre os modelos de solução ótima enumerativos, a enumeração pode se dar explicitamente ou implicitamente. Modelos de solução ótima de enumeração implícita diferenciam-se pelas particularidades de sua implementação e eficiência computacional. Os modelos desta categoria são os de programação inteira, *branch-and-bound* e programação dinâmica. Já os algoritmos eficientes particularizam-se como uma subclasse dos modelos otimizantes por gerarem soluções ótimas sem a necessidade de enumeração de todas as alternativas. Modelos de programação inteira podem ainda ser classificados quanto à forma de consideração do tempo por restrições

disjuntivas ou unidade de tempo discreto. O Quadro 9 apresenta os modelos de solução ótima subclassificados pela forma de enumeração.

Quadro 9 - Subclasses de modelos de solução ótima

Subclasses	Modelos
de enumeração explícita	enumeração total
de enumeração implícita	programação inteira, *branch-and-bound*, programação dinâmica
sem enumeração	Eficientes

Fonte: Pacheco e Santoro (1999, p. 12)

Os modelos heurísticos permitem que uma boa solução, não obrigatoriamente ótima, seja encontrada em tempo adequado (PACHECO; SANTORO, 1999). Atualmente na literatura os métodos heurísticos podem ser divididos em duas grandes classes: os métodos heurísticos construtivos e os melhorativos. Os métodos construtivos se caracterizam pelo fato de gerarem apenas uma solução, que será a solução final do problema. No caso dos métodos melhorativos, obtém-se uma solução inicial e, posteriormente, através de algum procedimento iterativo busca-se obter uma sequência das tarefas melhor que a atual, quanto à medida de desempenho adotada (NAGANO; BRANCO; MOCCELLIN, 2009).

Segundo Pinedo (2008) uma classe importante de métodos heurísticos compreende as rotinas *local search*, *simulated annealing* e *tabu search*, algoritmos genéticos, algoritmos *Ant Colony Optimization* (ACO). Muitos métodos heurísticos têm sido propostos para resolver problemas de *scheduling.*

2.2.3 *Job shop* com tempos de *setup* dependentes

Um dos problemas mais importantes da operação de *scheduling* abordado pela PO é o problema *job shop* (TARANTILIS; KIRANOUDIS, 2002). *Job shop scheduling* pode ser descrito como $\mathbf{J/C_{max}}$ de acordo com Graham *et al.*(1979), é um problema combinatorial, na maioria das vezes NP-completo.

A Figura 16 apresenta a hierarquia de complexidade das classes de problemas de *scheduling*.

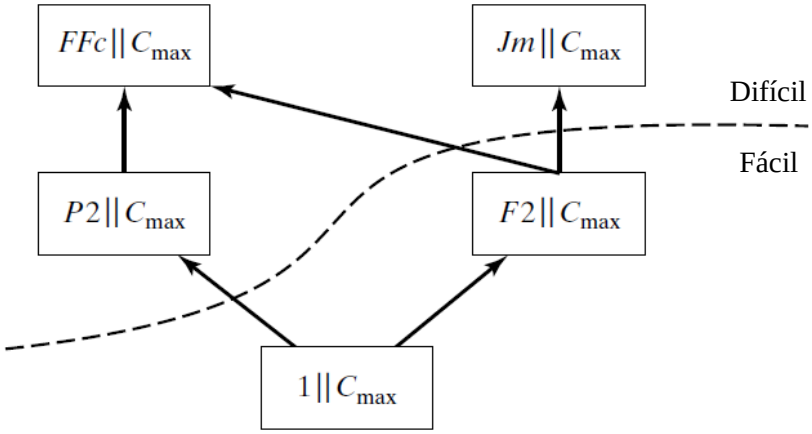

Figura 16 - Hierarquia de complexidade dos problemas de *scheduling*
Fonte: adaptado de Pinedo (2008)

Job shop scheduling é um processo de decisão que aloca **n** *jobs* para **m** máquinas, enquanto satisfaz dois tipos de restrição:uma delas é uma restrição tecnológica em que a sequência de máquinas que um *job* tem que visitar é fixa; a outra é uma restrição de capacidade em que uma máquina pode processar apenas um *job* de cada vez (CHUNG *et al.*, 2005). Naderi, Ghomi e Aminnayeri (2010) complementam descrevendo características adicionais como: quando uma operação é iniciada, ela não pode ser interrompida antes da conclusão, isto é, os *jobs* não podem ser interrompidos (*not preemptive*); não há tempo de transporte entre as máquinas, em outras palavras, quando uma operação de um *job* termina, a sua operação na máquina subsequente pode começar imediatamente; os *jobs* são independentes e estão disponíveis para o seu processo no tempo zero (0); as máquinas estão continuamente disponíveis.

Job shop scheduling (JSS) é um dos mais populares modelos de otimização da produção utilizados na prática (DEFERSHA; CHEN, 2010). Zhu e Wilhelm (2006) afirmam que a maioria dos problemas reais de *scheduling* envolvem tempos de *setup* (custos).

As operações com *setup* (tempo e/ou custo) foram consideradas por muito tempo insignificantes e, portanto, ignoradas, ou analisadas como parte do tempo de processamento. A consideração de tempos de *setup* em problemas *job shop* começou cerca de três décadas atrás, com a publicação do trabalho de Wilbrecht e Prescott (1969), que descobriram por meio de simulações que tempos de *setup* dependentes possuem um papel crítico no desempenho de uma operação *job shop* (ALLAHVERDI; GUPTA; ALDOWAISAN, 1999).

Em geral, o *setup* inclui o trabalho necessário para preparar uma máquina (ou processo) para produzir peças de um determinado tipo. Isso inclui obtenção e adequação de ferramentas, posicionamento do material no processo, limpeza, ajustes dos gabaritos e

inspeção do material. Existem dois tipos de problemas relacionados aos tempos de *setup*: no primeiro ele depende apenas do trabalho a ser processado, por isso é chamado de independente da sequência; no segundo, depende de ambos tanto do trabalho a ser processado e do trabalho imediatamente anterior, por isso é chamado dependente da sequência (ALLAHVERDI; GUPTA; ALDOWAISAN, 1999; NADERI; GHOMI; AMINNAYERI, 2010).

Os tempos de *setup* dependentes para *job shop scheduling* (SDST JSS) são definidos como **J/ STsd/ C_{max}** de acordo com Graham *et al.*(1979), com base nessa definição pode-se afirmar que o objetivo na solução de um JSS é determinar a ordem de processamento de todos os trabalhos em cada máquina que minimize o *makespan.*

Pode-se elencar vários artigos que analisaram pesquisas relacionadas à tempos de *setup*. Allahverdi, Gupta e Aldowaisan (1999) citaram 194 referências que tratam problemas de *setup*, sendo que a maioria foi publicada antes da década de 1990. Eles classificaram os problemas apresentados no artigo em *setups* dependentes ou independentes da sequência, bem como em lotes (*batch*) ou não (*non batch*). Esta classificação é aplicada a cada ambiente: máquina única, máquinas paralelas, *flow shop e job shop*. Do total dos trabalhos relacionados pelos autores, 13 abordam a relação entre ambientes *job shop* e tempos de *setup*, são eles: Wilbrecht e Prescott (1969), Hershauer (1970), Flynn (1984), Kim e Bobrowski (1994), Low (1995), O'Grady e Harrison (1998), Zhou e Egbelu (1989), Choi e Korkmaz (1997), Gupta (1982), Brucker e Thiele (1996), Ovacik e Uzsoy (1994), Patterson (1993), Sotoskov, Tautenhahn e Werner (1998).

Hershauer (1970), Flynn (1984), Kim e Bobrowski (1994) escreveram sobre *job shop* dinâmicos, eles como Wilbrecht e Prescott (1969) também usaram métodos de simulação para estudar os efeitos dos tempos de *setup* em sequências dependentes. Kim e Bobrowski (1994) examinaram as implicações dos tempos de *setup*, testando em diferentes ambientes regras de prioridade como *due date*, tempos de *setup* e estrutura de custo. Eles concluíram que os tempos de *setup* devem ser considerados na resolução de problemas de sequenciamento especialmente quando eles são dependentes da sequência.

Low (1995) utilizando simulação comparou o desempenho de seu algoritmo para tempos de *setup* com sequência dependente sob vários critérios com situações de tempos de *setup* com sequências independentes. O'Grady e Harrison (1998) propuseram uma regra de sequenciamento que prioriza trabalhos usando uma combinação linear das datas de entrega (*due date*), tempos de processamento e tempos de *setup* dependentes.

Zhou e Egbelu (1989) projetaram uma heurística para minimizar o *makespan* em um ambiente de produção flexível com tempos de *setup* dependentes. A heurística permitia interações com um operador humano por meio de uma *interface* gráfica. Choi e Korkmaz (1997) formularam o mesmo problema com programação inteira mista e desenvolveram uma heurística polinomial que gerou melhor desempenho ao trabalho de Zhou e Egbelu (1989).

Gupta (1982) desenvolveu um algoritmo com base em *branch-and-bound* para minimizar os custos em ambientes *job shop* com tempos de *setup* dependentes. Brucker e Thiele (1996) também desenvolveram um algoritmo *branch-and-bound* para o problema de sequenciamento em ambiente *job shop* com tempos de *setup* dependentes.

Ovacik e Uzsoy (1994) desenvolveram técnicas de programação que usam informações em tempo real do chão de fábrica. Após a comparação dessas técnicas com as tradicionais regras de prioridade em um ambiente *job shop*, eles identificaram que o desempenho era melhor em ambientes com restrições de recursos e fluxo. Patterson (1993) analisou tempos de *setup* dependentes com restrições de recursos.

Sotoskov, Tautenhahn e Werner (1998) desenvolveram uma heurística construtiva para minimizar várias funções objetivo considerando lotes e tempos de *setup* independentes.

Outro artigo que analisou pesquisas relacionadas à tempos de *setup* é o escrito por Zhu e Wilhelm (2006), nele os autores citam 182 referências que abordam dimensionamento de lotes com tempos de *setup* com sequência dependente. Eles organizaram a pesquisa nos ambientes: máquina única, máquinas paralelas, *flow shop* e *job shop*. Do total dos trabalhos relacionados pelos autores, 17 abordam a relação entre ambientes *job shop* e tempos de *setup* dependentes. Como alguns trabalhos foram também referenciados por Allahverdi; Gupta; Aldowaisan (1999) pode-se identificar 11 novas pesquisas relacionadas ao problema em questão, são eles: Candido, Khator eBarcia (1998), Hertz e Widmer (1996), Cheung e Zhou (2002), Choi e Choi (2002), Sun, Yee e Hwang (2003), Artigues e Roubellat (2002), Zoghby, Barnes e Hasenbein (2005), Kim e Bobrowski (1997), Luh *et al.* (1998), Mason, Fowler e Carlyle (2002) e Mason *et al.*(2005).

Allahverdi *et al.* (2008) escreveram a continuação da pesquisa anterior (ALLAHVERDI; GUPTA; ALDOWAISAN, 1999) avaliando mais de 300 artigos que foram publicados entre 1999-2006. Desde a publicação do trabalho, em 1999, os autores afirmam que houve um aumento no interesse sobre problemas de *scheduling* com tempos ou custos de *setup*, somando uma média de 40 *papers* por ano sobre o assunto. Novamente classificam os problemas de acordo com o ambiente (*flow shop, no-wait flow shop, flexible flow shop, job*

shop, open shop, máquina única e máquinas paralelas) e ainda em *setups* dependentes ou independentes da sequência, bem como em lotes (*batch*) ou não (*non batch*). São 10 artigos referentes a problemas *job shop*: Sun e Noble (1999), Focacci, Laborie e Nuijten (2000), Ballicu, Giua e Seatzu (2002), Artigues e Buscaylet (2003), Artigues, Belmokhtar e Feillet (2004), Artigues, Buscaylete Feillet (2005), Balas, Simonetti e Vazacopoulos (2005), Tahar *et al.* (2005), Aldakhilallah e Ramesh (2001), Low, Hsu e Huang (2004).

O Quadro 10 apresenta um resumo dos trabalhos, critérios e métodos de solução apresentados nos artigos de Zhu e Wilhelm (2006) e Allahverdi *et al.* (2008), relacionados a ambientes *job shop* com tempos de *setup* dependentes.

Quadro 10 - Literatura sobre *scheduling* em ambiente *job shop* com *setup* dependente

Referências	Critérios	Método de solução
Artigues e Roubellat (2002)	$\mathbf{J_m/r_j, s_{ij}/L_{max}}$ (problema de inserção)	Algoritmo de inserção
Candido, Khator e Barcia (1998)	$\mathbf{J_m/s_{ij}/\mathfrak{R}}$	Algoritmo genético (GA) – baseado em heurística
Cheung e Zhou (2002)	$\mathbf{J_m/s_{ijk}/C_{max}}$	GA híbrido e heurística com regra SPT
Choi e Choi (2002)	$\mathbf{J_m/s_{ijk}/C_{max}}$	Programação Linear Inteira Mista (MIP) e *local search*
Hertz e Widmer (1996)	$\mathbf{J_m/s_{ijk}/C_{max}}$	*Tabu search* (TS)
Kim e Bobrowski (1997)	$\mathbf{J_m/\tilde{s}_{ijk}\|\gamma}$	Regras de *dispatching*
Luh *et al.* (1998)	$\mathbf{FJ_c/block, bs_{ijk}, r_j} / \sum w'_j E_j^2 + \sum w''_j T_j^2$	MIP e hibrido: relaxamento Langragiano, Programação dinâmica (DP) + heurística
Mason *et al.*(2005)	$\mathbf{FJ_c/recrc, r_j, s_{ij}} / \sum \mathbf{w_j T_j}$	MIP, heurística, regras de *dispatching*
Mason, Fowler e Carlyle (2002)	$\mathbf{FJ_c/recrc, r_j, s_{ij}} / \sum \mathbf{w_j T_j}$	Alteração na heurística do gargalo
Sun, Yee e Hwang (2003)	$\mathbf{J_m/recrc, s_{ijk}/C_{max}}$	GA híbrido e heurística
Zoghby, Barnes e Hasenbein (2005)	$\mathbf{J_m/recrc, s_{ij}/\gamma}$	Meta-heurística e gráfico disjuntivo
Sun e Noble (1999)	$\mathbf{J_m/s_{ijk}/\sum w_j T_j^2\ (r_j)}$	Formulação Matemática
Focacci, Laborie e Nuijten (2000)	$\mathbf{J_m/s_{ijk}/C_{max}}$	*Branch-and-bound*
Ballicu, Giua e Seatzu (2002)	$\mathbf{J_m/s_{ijk}/C_{max}}$	Programação Inteira Mista

Continuação do Quadro 10 - Literatura sobre *scheduling* em ambiente *job shop* com *setup* dependente

Artigues e Buscaylet (2003)	$\mathbf{J_m/s_{ijk}/C_{max}}$	*Tabu search*
Artigues, Belmokhtar e Feillet (2004)	$\mathbf{J_m/s_{ijk}/C_{max}}$	*Branch-and-bound*
Artigues, Buscaylet e Feillet (2005)	$\mathbf{J_m/s_{ijk}/C_{max}}$	Integração de *tabu search* e heurísticas
Balas, Simonetti e Vazacopoulos (2005)	$\mathbf{J_m/s_{ijk,}r_j/C_{max}}$	Gargalo e programação dinâmica
Tahar *et al.* (2005)	$\mathbf{\sum T_j}$ **($\mathbf{r_j}$, restrições de precedência, *job shop* hibrido)**	*Ant colony*
Aldakhilallah e Ramesh (2001)	***job shop* reentrante**	Heurística Construtiva
Low, Hsu e Huang (2004)	$\mathbf{J_m/s_{ijk}/C_{max}}$	Programação Inteira e gráfico disjuntivo

Fonte: adaptado de Zhu e Wilhelm (2006) e Allahverdi *et al.* (2008)

Pode-se verificar, com base nos artigos relacionados acima, que poucos trabalhos usaram métodos de solução ótima em problemas de sequenciamento em ambientes *job shop* considerando os tempos de *setup* dependentes. Dos 34 trabalhos relacionados, nove usam esse tipo de método de solução, mas nenhum usa os planos de corte de Gomory.

2.2.4 Modelos de solução para problemas *Job shop Scheduling*

Pacheco e Santoro (1999) propõem uma classificação hierarquizada para os problemas de *job shop scheduling*, os modelos foram distribuídos inicialmente em duas classes: modelos de solução ótima e heurísticos. De acordo com essa classificação, verifica-se então, a partir desse tópico, os métodos de solução de problemas de *scheduling* em ambientes *job shop.*

2.2.4.1 Métodos de Solução Ótima

Dentre os modelos de solução ótima identificam-se a enumeração total, os algoritmos eficientes, a programação inteira, *branch-and-bound* e a programação dinâmica. A Figura 17 apresenta a hierarquia dos modelos de solução ótima.Os retângulos sombreados com cantos arredondados contêm os modelos, os retângulos sombreados com bordas pontilhadas

representam detalhes relevantes dos modelos e os retângulos sem sombreamento representam as classes de modelos.

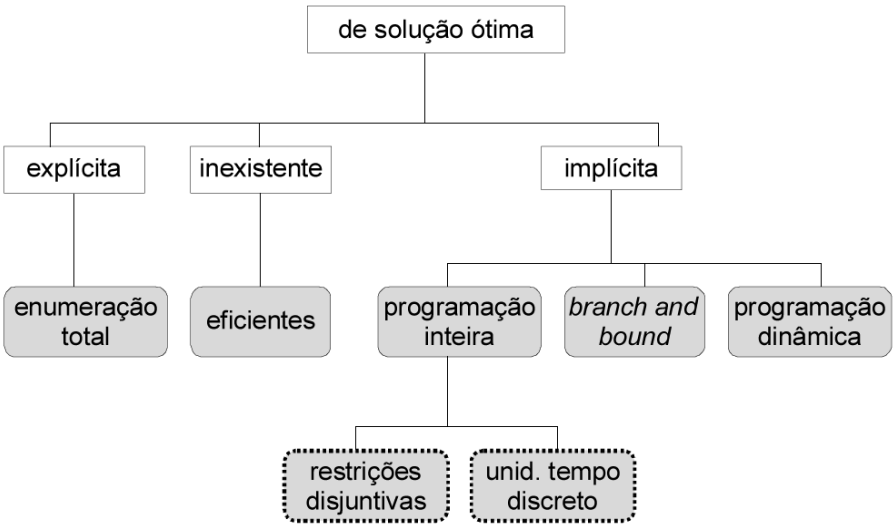

Figura 17 - Classificação hierarquizada de modelos de solução ótima para o problema *job shop*
Fonte: Adaptado de Pacheco e Santoro (1999)

Enumeração total caracteriza-se pelo uso de técnicas genéricas para a solução de problemas de *scheduling* ou de outro problema combinatorial, diferindo pela forma como são definidas a função objetivo, as restrições e os parâmetros. Os algoritmos enumerativos não utilizam características específicas do problema para direcionar a busca. No entanto, esta forma de abordar o problema não permitirá encontrar, para todos os casos, a solução ótima em tempo útil, pois em geral, o número de soluções pode ser muito elevado. Para um problema com **n** tarefas e **m** máquinas haverá uma possibilidade de $(n!)^m$ soluções. Assim, a enumeração completa é um método cuja aplicabilidade fica restrita a casos especiais de problemas de *scheduling* de dimensão reduzida (BAKER, 1974; PACHECO; SANTORO, 1999; BAKER; TRIETSCH, 2009).

Os algoritmos eficientes particularizam-se como uma subclasse dos modelos otimizantes por gerarem soluções ótimas sem a necessidade de enumeração de todas as alternativas. Infelizmente, poucos são os problemas cuja solução pode ser encontrada por algoritmos eficientes e, mesmo os existentes, possuem hipóteses pouco aderentes aos problemas normalmente encontrados na realidade (PACHECO; SANTORO, 1999).

Já a programação inteira pesquisa caminhos para resolver problemas de otimização com variáveis discretas ou inteiras. Tais variáveis são usadas para modelo indivisíveis, e variáveis como 0/1 são usadas para representar sim ou não em decisões de compra, investimento, contrações, e assim por diante. Esses problemas surgem em todas as situações

do cotidiano, por exemplo, no desenvolvimento dos horários dos trens ou horários de aeronaves, planejando o sequenciamento do trabalho de uma linha de produção ou da equipe de manutenção, entre outras situações (WOLSEY, 1998).

De acordo com Jünger *et al.* (2010) o campo da programação de inteira alcançou grande sucesso na área acadêmica e no mundos dos negócios. São publicados anualmente centenas de artigos, realizam-se várias conferências internacionais e *softwares* são desenvolvidos para resolver problemas de programação inteira. As áreas de aplicação incluem logística e *supply chain*, telecomunicações, finanças, manufatura e muitos outros.

Segundo Lachtermacher (2009) o algoritmo *branch-and-bound* é o procedimento mais utilizado atualmente na resolução de problemas de programação linear inteira ou inteira mista. O método divide o conjunto de soluções viáveis em subconjuntos sem intersecção entre si, calculando os limites superior e inferior para cada subconjunto, e então eliminando os subconjuntos de acordo com as regras pré-estabelecidas. Segundo Reis (1996) há três aspectos básicos neste método:

- O procedimento de *branch* (ramificar): divisão de um problema grande em dois ou mais subproblemas.
- O procedimento de *bound* (limitar): calcula um limite inferior da solução ótima para cada subproblema.
- A estratégia de busca usada: relaciona-se com os subproblemas que são escolhidos para tomar em consideração em cada passo.

Pinedo (2008) afirma que a simplicidade do princípio básico do *branch-and-bound* e sua eficiência computacional parecem ser razões básicas da sua popularidade.

Programação dinâmica é uma técnica matemática que utilizada para criar uma sequência de decisões inter-relacionadas. Ela fornece um procedimento sistemático para determinar a combinação de decisões ótimas, é um processo de decisão em vários estágios: em cada estágio deve ser tomada uma decisão que tem impacto nas decisões a serem tomadas em estágios subsequentes (BAKER; TRIETSCH, 2009; HILLIER; LIEBERMAN, 2010). Pinedo (2008) afirma que a programação dinâmica é basicamente um esquema de enumeração completa que por meio de uma abordagem de dividir para conquistar, busca minimizar a quantidade de processamento necessária.

2.2.4.1.1 Planos de Corte: Gomory

O ano 2008 marcou o 50° aniversário do aparecimento do artigo publicado por Gomory, algoritmo usado como base dessa pesquisa. Pode-se contextualizá-lo a partir da teoria da programação inteira. Pode-se considerar como o primeiro pesquisador da área, George Dantzig, que em 1947, criou o algoritmo simplex que foi publicado em 1951. Este artigo descreveu um método finito para otimizar uma função objetivo linear sujeita a um conjunto finito de restrições lineares. Um grande número de problemas podia ser modelado usando funções lineares e variáveis inteiras, mas não se conhecia nenhum método para resolução de problemas de programação inteira mista (JÜNGER *et al.*, 2010).

Balinski (1965) afirma que em certas circunstâncias é possível predizer que um programa linear necessariamente deverá ter uma solução inteira. Identificam-se problemas matemáticos com essa característica aqueles relacionados à atribuição, transporte, e problemas de fluxo de rede. Por exemplo, George Dantzig aponta em 1949 que o problema de transporte (e consequentemente esta classe de problemas) sempre deverá ter soluções inteiras, dados os valores inteiros relacionados a demanda e recursos. Porém, deve-se atentar ao fato de existirem outros problemas (matemáticos) que têm esta propriedade, de áreas diversas e diferentes das relacionadas anteriormente (atribuição, transporte, e problemas de fluxo de rede).

A abordagem básica mais comum dos métodos é aquela de deduzir sucessivamente as restrições lineares e seus requisitos dos problemas de programação inteira, até que um novo problema linear seja obtido de forma a satisfazer os requisitos inteiros. A idéia inicial da nova geração de restrições surgiu com Dantzig, Fulkerson, e Johnson (1954), no trabalho desenvolvido para resolver o problema do caixeiro viajante, e em seguida, por Markowitz e Manne (1957). Em ambos os casos a idéia foi usada de uma *ad hoc* em problemas específicos.

Em 1958, Ralph Gomory publicou um pequeno artigo que descrevia como, o algoritmo simplex de Dantzig podia ser adaptado para gerar um algoritmo finito de forma a achar uma solução ótima inteira para um problema de programação linear (JOHNSON, 2005).

Esse trabalho teve origem em uma apresentação de um modelo de programação linear no *Naval Research* em Washington, onde um dos apresentadores destacou que seria interessante ter um número inteiro em lugar de 1,3 aviões. Gomory passou a estudar um método que produzisse resultados inteiros para qualquer problema que necessitasse desse tipo de abordagem. Na tarde do oitavo dia, pensou que se tivesse que obter uma solução inteira tendo como base um problema de programação linear inteira com coeficientes inteiros na

função objetivo, o primeiro passo seria obter o valor ótimo da função objetivo a ser maximizada através de programação linear, por exemplo, 7.14 e então concluiu que o valor máximo inteiro seria sete. Mas, porque sete? Como ele estava trabalhando com equações que tinham coeficientes inteiros e somente com variáveis inteiras, não demorou a identificar que dois passos eram fundamentais. Em primeiro lugar o valor ótimo dado por programação linear é um limitante superior para o valor ótimo de uma solução inteira. Em segundo lugar, se os coeficientes da função objetivo são inteiros, então é válido adicionar a desigualdade linear impondo que a função objetivo deverá ser menor que 7.14, ou seja, sete (JÜNGER *et al.*, 2010).

Esta desigualdade foi a base para Gomory desenvolver o método dos cortes fracionários, ele mostrou em seu artigo que a tabela simplex podia ser usada para gerar novas desigualdades que eram válidas para todas as soluções satisfazendo a integralidade das restrições. O estudo destas desigualdades, chamados cortes, rapidamente tornou-se uma área importante de pesquisa por razões teóricas e também pela promessa futura de desenvolvimento de ferramentas computacionais.

A partir de Gomory (1958) e Nemhauser e Wolsey (1999) descreve-se o modelo geral dos planos de corte.

Considere

(PI) Max $\{x_0: (x_0, x) \in S^e\}$, onde $S^0=\{x_0 \in Z^1, x \in Z_+^n: x_0 - cx = 0, Ax = b\}$.

Supõe-se que uma base ótima para o relaxamento de programação linear foi obtida, então PI pode ser escrito como

$$\text{Max } x_0$$

$$x_{Bi} + \sum_{j \in H} \bar{a}_{ij} x_j = \bar{a}_{i0} \quad \text{para } i = 0, 1, ..., m$$

$$x_{B0} \in Z,\ x_{Bi} \in Z_+^1 \text{ para } i = 1, ..., m,\ x_j \in Z_+^1 \text{ para } j \in H,$$

Onde $x_0 = x_{B0}, x_{Bi}$ para $i = 1, ..., m$ são as variáveis básicas e onde x_j para $j \in H \subset N = \{1, ..., n\}$ são as variáveis não básicas. Desde a base primal e dual possível, tem-se $\bar{a}_{i0} \geq 0$ para $i = 1, ..., m$ e $\bar{a}_{0j} \geq 0$ para $j \in H$.

Se $\bar{a}_{i0} \notin Z^1$, então

$$\sum_{j \in H} f_{ij} x_j = f_{i0} + x_{n+1}, \quad x_{n+1} \in Z_+^1,$$

onde $f_{ij} = \bar{a}_{ij} - |\bar{a}_{ij}|$ para $j \in H$ e $f_{i0} = \bar{a}_{i0} - |\bar{a}_{i0}|$, é a igualdade válida para S^0.

Exemplo

$$\text{Max } x_0$$

$$\begin{aligned} x_0 - 7x_1 - 2x_2 \qquad\qquad\qquad &= 0 \\ -x_1 + 2x_2 + x_3 \qquad\qquad &= 4 \\ 5x_1 + x_2 \qquad + x_4 \qquad &= 20 \\ -2x_1 - 2x_2 \qquad\qquad + x_5 &= -7 \end{aligned}$$

$x_0 \in Z^1, x_j \in Z^1_+$ para j = 1, ..., 5.

Uma solução ótima para o problema de programação linear relaxado é

$$\begin{aligned} x_0 \qquad\qquad + \frac{3}{11}x_3 + \frac{16}{11}x_4 \qquad &= \frac{332}{11} \\ x_1 \qquad - \frac{1}{11}x_3 + \frac{2}{11}x_4 \qquad &= \frac{36}{11} \\ x_2 - \frac{5}{11}x_3 + \frac{1}{11}x_4 \qquad &= \frac{40}{11} \\ \frac{8}{11}x_3 + \frac{6}{11}x_4 + x_5 &= \frac{75}{11} \end{aligned}$$

Onde $x_3 = x_4 = 0$.

Gerando o corte fracionário obtém-se

$$\frac{3}{11}x_3 + \frac{5}{11}x_4 = \frac{2}{11} + x_6,\ x_6 \in Z^1_+$$

Em termos das variáveis originais, o corte é $2x_1 + x_2 \leq 10$.

Segundo Arenales *et al.* (2007, p. 254) o algoritmo de Gomory pode ser descrito em quatro passos:

Passo 1 (Inicialização): Faça k = 0 e PL^0= PL, em que PL é a relaxação linear do problema P.

Passo 2 (Reotimização): Resolva o problema linear, pelo método simplex.

$$\bar{z}^k = \max \{\mathbf{c}^{\mathbf{T}}\mathbf{x}: \mathbf{x} \in PL^k\}$$

Passo 3 (Otimalidade): Se a solução for inteira, então é uma solução ótima de P, caso contrário vá para o Passo 2.

Passo 4 (Corte):Escolha uma linha *i* com $f_{io} > 0$, construa o corte de Gomory e insira-o no fim do quadro ótimo de PL^k. Faça k = k + 1 e vá para o Passo 2.

Em 1960, o corte foi estendido para problemas de programação inteira mista, que gerou um corte fracionário mais forte para programação inteira. Gilmore e Gomory (1961;

1963) desenvolveram uma abordagem de programação linear para problemas de corte de estoque com o objetivo de minimizar o custo. O problema era cortar a partir de um número ilimitado de barras disponíveis em estoque de tamanho L_k, N_i peças de tamanho l_i de acordo com a demanda de cada peça. O método proposto nesses trabalhos pode ser apresentado, resumidamente, da seguinte maneira:

- Gera-se um subconjunto de padrões de corte, cuja diagonal é o resultado de $L/l_{i;}$
- Resolve-se o problema de minimização do custo até a otimalidade;
- Resolve-se o problema da mochila;
- Adiciona-se a nova coluna gerada aos padrões de corte (última coluna da matriz) e volta-se a resolver o problema de minimização do custo até a otimalidade.

Este método possibilita trabalhar apenas com o subconjunto de padrões que tem uma propriedade em comum: todas as variáveis associadas às colunas geradas são candidatas a estar na base da solução, ou seja, ao invés de se trabalhar com todo o universo de colunas, trabalha-se somente com as melhores opções (DAL GALLO, 2008).

Segundo Arenales *et al.* (2007) os problemas de corte consistem em cortar peças menores (itens), em tamanhos e quantidades variadas, a partir de peças maiores (objetos). Surge, então um problema de otimização para minimizar a perda de material dos objetos. O problema de corte pode ser unidimensional, isto é, apenas uma dimensão é importante no processo de corte, por exemplo, bobinas de papel e barras de aço; ou bidimensional, nesse caso duas dimensões são importantes para o processo de corte, por exemplo, placas de madeira.

Apresenta-se a formulação matemática de um problema de corte unidimensional, pois essa modelagem pode ser estendida para problemas com mais dimensões. Nesse problema deseja-se cortar barras disponíveis de uma tamanho padrão **L** para a produção de **m** itens com tamanhos $l_1, l_2,..., l_m$ em várias quantidades $b_1, b_2,..., b_m$, respectivamente. A maneira como se corta uma barra define o chamado **padrão de corte**, para cada padrão de corte **j** (j = 1, 2,..) associa-se um vetor m-dimensional $\mathbf{a_j} = (a_{1j}, a_{2j},...,a_{mj})$, em que $\mathbf{a_{ij}}$ fornece o número de itens do tipo **i** no padrão de corte **j** (ARENALES *et al.*, 2007; DAL GALLO, 2008).

Um vetor $a = (a_1, a_2, ..., a_m)$ representa um padrão de corte se e somente se o seguinte sistema é satisfeito:

$$l_1a_1 + l_2a_2 + ... + l_ma_m \leq L$$

$$a_1 \geq 0,\ a_2 \geq 0,\ ...,\ a_m \geq 0 \text{ e inteiros.}$$

De acordo com Arenales *et al.* (2007) uma vez definidos os padrões de corte, o problema consiste em determinar quantas barras devem ser cortadas de acordo com cada padrão, de modo que a demanda de cada item seja atendida, utilizando-se o menor número possível de barras. Define-se $\mathbf{x_j}$ como o número de barras cortadas conforme o padrão **j**. O problema de corte pode ser formulado por:

$$\text{Minimizar } f\ (x_1, x_2, \dots, x_n) = x_1 + x_2 + \dots + x_n$$

$$\begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} x_1 + \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} x_2 + \dots + \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} x_n = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

$$x_j \geq 0, j = 1, \dots, n.$$

O problema da mochila é considerado um clássico e envolve a escolha de itens a serem colocados em uma ou mais mochilas de forma a maximizar uma função de utilidade. Considere **n** projetos e um capital **b** para investimento. O projeto **j** tem um custo $\mathbf{a_j}$ e um retorno esperado $\mathbf{p_j}$. O problema consiste em selecionar os projetos que maximizem o retorno total esperado sem ultrapassar o limite de capital (**b**) (PINEDO, 2008; ARENALES *et al.*, 2007).

As variáveis:

$$x_j \begin{cases} 1\ se\ o\ projeto\ j\ for\ selecionado \\ 0\ caso\ contrário \end{cases}$$

O problema é formulado por:

$$\boldsymbol{Max} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{p_j x_j}$$

$$\sum_{j=1}^{n} a_j x_j \leq b$$

$$x_j \geq 0 \text{ e inteiro}, \qquad j = 1, .., n$$

Nas décadas de 70 e 80 não foram publicados artigos reportando experimentos computacionais abrangentes com os cortes fracionários (FONSECA, 2007). A comunidade científica, no início da década de 90, acreditava que para resolver problemas de programação inteira de tamanho significativo tinha-se que explorar a estrutura do problema combinatório. Os cortes de Gomory geraram uma teoria elegante, mas não tinham utilidade na prática,

porque eram cortes genéricos e não exploravam a estrutura (CORNUÉJOLS, 2007).

Balas *et al.* (1996) demonstraram que os cortes de Gomory para problemas mistos com variáveis binárias são válidos em todos os nós da árvore de enumeração, permitindo que cortes gerados em nós distintos possam ser compartilhados. Os autores destacam três razões para o bom funcionamento dos cortes de Gomory: resolvedores de programação linear mais robustos, adição de todos os cortes do *tableau* e uso na estrutura *branch-and-cut* (em lugar de um enfoque de cortes puros).

Este tipo de resultado, certamente influenciou na inclusão de cortes de Gomory nos pacotes comerciais de otimização como CPLEX, XPRESS e LINDO, e também para renovar o interesse de pesquisadores por cortes baseados em Gomory (FONSECA, 2007).

2.2.4.2 Métodos Heurísticos

Segundo Reis (1996) em face ao obstáculo da complexidade passou-se a dar maior atenção e importância a métodos aproximados, do tipo heurístico. Dentre os modelos heurísticos identificam-se o *beam search*, sistemas especialistas, *tabu search*, algoritmos genéticos, *simulated annealing*, redes neurais, programação de gargalos. Na classificação de Pacheco e Santoro (1999) os heurísticos foram subdivididos em métodos de passo único, métodos de busca.

- Métodos de busca procuram gerar diversas soluções, escolhendo a melhor delas.
- Métodos de passo único fornecem uma solução com apenas uma execução do algoritmo.

As Figuras 18 e 19 apresentam a hierarquia dos modelos heurísticos de acordo com a classificação de Pacheco e Santoro (1999). Os retângulos sombreados com cantos arredondados contêm os modelos, os retângulos sombreados com bordas pontilhadas representam detalhes relevantes dos modelos e os retângulos sem sombreamento representam as classes de modelos. A Figura 18 apresenta a hierarquia de classificação dos modelos heurísticos de busca.

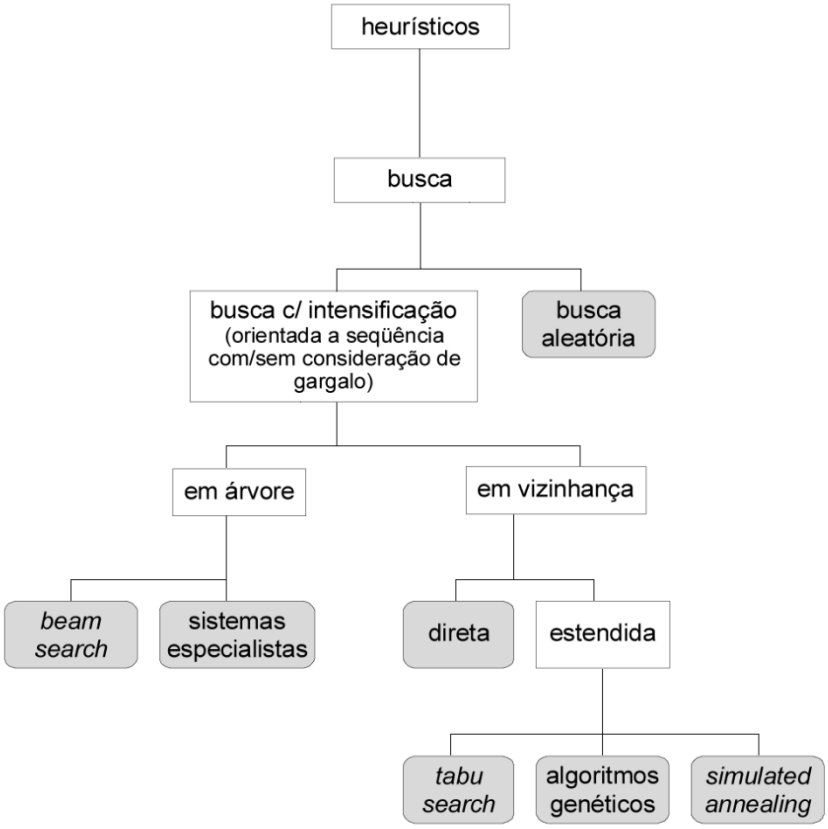

Figura 18 - Classificação hierarquizada dos modelos heurísticos de busca
Fonte: Adaptado de Pacheco e Santoro (1999)

O *beam search* é uma heurística do *branch-and-bound* que não necessariamente avalia a árvore completa. Essa abordagem sacrifica a garantia de obtenção de um resultado ótimo em função dos ganhos de velocidade. Para cada nível da árvore, apenas um número limitado de nós são selecionados para ramificação, os outros nós são definitivamente descartados. Os nós selecionados para a ramificação são chamados de *beam width.* (AZIZOGLU; WEBSTER, 1997; PINEDO, 2008).

Se o conhecimento é representado na forma de regras *IF-THEN*, então o problema poderá ser codificado usando um sistema especialista. Esses sistemas possuem um mecanismo de inferência que é capaz de fazer o encadeamento para frente ou encadeamento para trás das regras, a fim de obter uma solução viável. Sistemas especialistas oferecem uma nova abordagem para o desenvolvimento de bons sequenciamentos em tempo hábil (BIEGEL; WINK, 1989; PINEDO, 2008).

Segundo Pacheco e Santoro (1999) os métodos de busca em vizinhança que utilizam mecanismos de diversificação são chamados de métodos de busca estendida, por permitirem que a busca se estenda explorando alternativas vizinhas melhores do que a solução atual. Nesta categoria encontram-se os modelos denominados *tabu search*, *simulated annealing* e algoritmos genéticos.

Simulated annealing e *Tabu search* são considerados algoritmos de melhoramento, porque a partir de uma solução inicial, avançam para outra solução (melhor que a anterior) na sua vizinhança até que se satisfaça um determinado critério de parada. A diferença entre esses modelos está no critério de aceitação-rejeição, no *simulated annealing* o critério é baseado em um processo probabilístico enquanto que no *tabu search* é baseado em um processo determinístico (PINEDO, 2008).

O *simulated annealing* é um processo de pesquisa que tem sua origem nos campos da ciência dos materiais e da física. Foi desenvolvido inicialmente como um modelo de simulação para descrever o processo físico para fundir um metal, onde este é aquecido a uma temperatura elevada e em seguida é resfriado lentamente, de modo que o produto final seja uma massa homogênea. De forma equivalente, o processo de otimização é realizado por níveis, simulando os níveis de temperatura no resfriamento. O algoritmo substitui a solução atual por uma solução próxima (na sua vizinhança no espaço de soluções), escolhida de acordo com uma função objetivo e com uma variável *T* (Temperatura, por analogia).Cada ponto gerado é aceito ou rejeitado de acordo com certa probabilidade. Esta probabilidade de aceitação decresce de acordo com o nível do processo, ou equivalentemente, de acordo com a temperatura (HAESER; GOMES–RUGGIERO, 2008).

O *Tabu search*, partindo de uma solução inicial, move-se, a cada iteração, para a melhor solução na vizinhança não aceitando movimentos que levem a soluções já visitadas. Essa identificação é possível por permanecerem armazenadas em uma lista *tabu*. A lista permanece na memória guardando soluções já visitadas (*tabu*) durante um determinado espaço de tempo ou certo número de iterações (prazo *tabu*). Como resultado final é esperado que se encontre um ótimo local ou a solução mais próxima deste (PINEDO, 2008).

Segundo Pinedo (2008) algoritmos genéticos são mais gerais e abstratos que *simulated annealing* e *tabu search*, que podem, de certa forma, ser vistos como casos especiais de algoritmos genéticos. Normalmente, um GA cria uma lista de soluções promissoras em cada etapa, e as iterações geram os melhores resultados pesquisando um tipo especial de vizinho. Em vez de definir um vizinho, alterando uma única sequência, a GA combina duas sequências existentes, selecionando algumas características de um e o restante do outro (BAKER; TRIETSCH, 2009).

A Figura 19 apresenta a hierarquia de classificação dos modelos heurísticos de passo único.

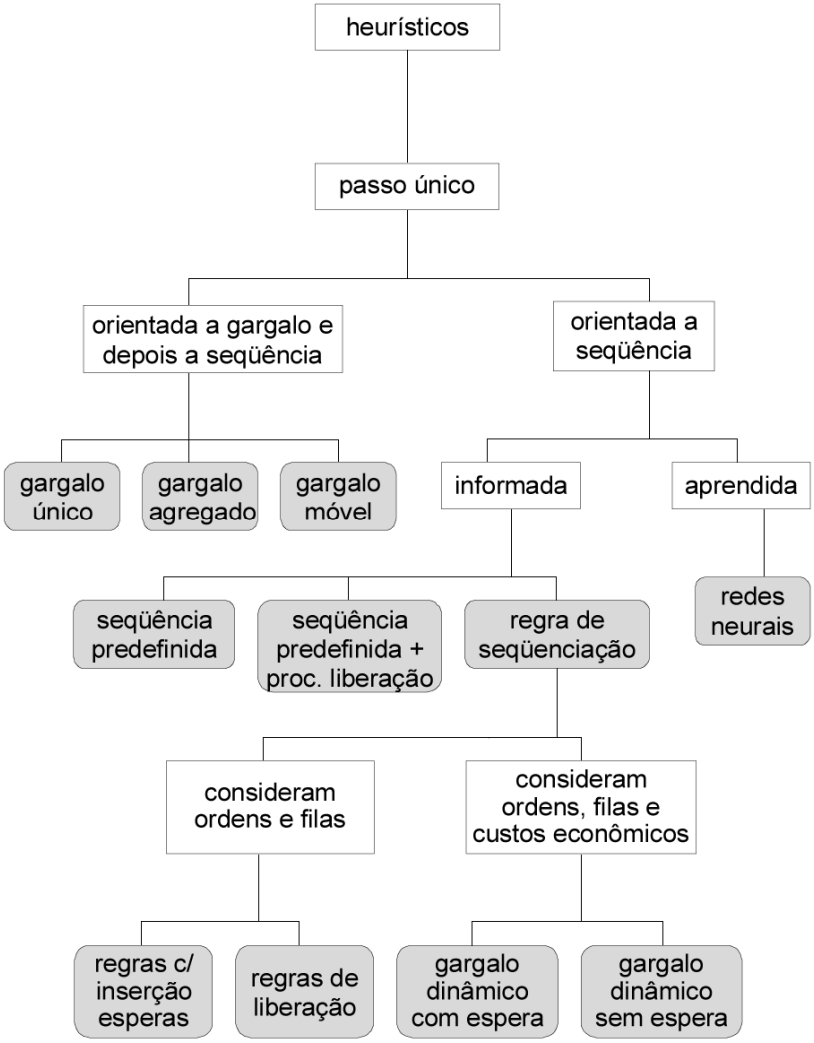

Figura 19 - Classificação hierarquizada dos modelos heurísticos de passo único
Fonte: Adaptado de Pacheco e Santoro (1999)

Segundo Pacheco e Santoro (1999) os modelos heurísticos de passo único orientados à sequência podem gerar boas soluções de duas formas: aprendida pelo modelo ou informada. Na primeira categoria (aprendida) encontram-se os modelos de redes neurais, nos quais o modelo **aprende** a melhor forma de resolver o problema de *scheduling* por meio de um conjunto-exemplo de problemas que **calibra** a malha de neurônios. Na segunda categoria (informada), o procedimento de resolução do problema (ou lógica) é incorporado na modelagem.

Já as heurísticas de passo único orientadas a gargalo se caracterizam por procurar resolver o problema de *scheduling* inicialmente nas máquinas mais críticas com respeito à limitação de capacidade. Os modelos existentes se diferenciam pela quantidade e pela forma de determinação dos gargalos. A heurística *Shifting Bottleneck* sequencia cada máquina separadamente, resolvendo um problema de escalonamento de uma única máquina a cada

passo. Cada vez que uma nova máquina é sequenciada, as máquinas já sequenciadas anteriormente são re-sequenciadas, levando em conta o novo estado do problema (PINEDO, 2008).

Os métodos heurísticos descritos apresentam vantagens: são aplicáveis a uma larga quantidade de problemas; podem ser usados em combinação com outros métodos; tornam os modelos mais próximos do problema real, além de serem de fácil implementação. Um aspecto que, no entanto, penaliza fortemente estes algoritmos é o de não serem eficientes. Mesmo abandonando-se o objetivo de encontrar uma solução ótima e contentando-se com uma solução satisfatória para um problema, a obtenção desta solução pode ser demorada. Em outras palavras, não serão apropriados para *scheduling* reativo, isto é, *scheduling* com um horizonte temporal curto (REIS, 1996).

A abordagem de resolução de um problema por meio da PO envolve várias fases, por exemplo, definição do problema, construção do modelo e outras que serão apresentadas no capítulo subsequente.

3 O MÉTODO DE PESQUISA

De acordo com o que Chizzotti (2005) argumenta, a pesquisa científica estuda o mundo em que o homem vive e o próprio homem. Para isso, defronta-se com todas as forças da natureza e de si mesmo, reúne todas as energias de sua capacidade criadora, organiza todas as possibilidades da sua ação e seleciona as melhores técnicas e instrumentos para descobrir algo que transforme os horizontes de sua vida. Mudar o mundo, criar objetos e concepções, encontrar explicações e confirmar previsões, são fins subjacentes a todo esforço de pesquisa. Seguindo o seu raciocínio, Chizzotti (2005, p. 20) defende ainda que

> (...) como a pesquisa é um esforço durável de busca, análise e síntese, impõe uma disciplina que responda ao volume físico e mental de trabalho por um período duradouro e exige a adoção de métodos condizentes com a complexidade das questões que deverão ser resolvidas.

Sendo assim, a pesquisa deve ser planejada com extremo rigor. A existência de um rigor metodológico visa assegurar a realização do estudo. Percebendo-se a necessidade deste rigor metodológico, a pesquisa deverá fundamentar-se em métodos adequados ao tipo de estudo que vislumbra. Segundo Cooper e Schindler (2003, p. 33) “a boa pesquisa segue os padrões do método científico”.

3.1 Tipo de Pesquisa

De acordo com as referências bibliográficas que fundamentam esse capítulo, o presente trabalho se caracteriza como sendo uma pesquisa aplicada, com o desenvolvimento de um modelo. O trabalho foi organizado em duas etapas: pesquisa bibliográfica e pesquisa experimental.

De acordo com Gil (1999, p. 65) “a pesquisa bibliográfica é desenvolvida a partir de material já elaborado, constituído principalmente de livros e artigos científicos”. Ela representa uma base fundamental para o desenvolvimento deste trabalho.

A pesquisa experimental foi realizada a partir do desenvolvimento do modelo e avaliação de sua aplicabilidade. Segundo Chizzotti (2005) este método consiste em submeter um fato à experimentação em condições de controle e apreciá-lo coerentemente e rigorosamente, mensurando as ocorrências e suas exceções. Devendo reconhecer como científico, somente o que for passível de apreensão em condições de controle, legitimados pela experimentação e validados pela mensuração.

Cooper e Schindler (2003) definem modelo como a representação de um sistema construído para estudar algum aspecto daquele sistema ou o sistema como um todo. O modelo difere da teoria, pois um tem o papel de explicação e o outro de representação. Podem ser usados para propósitos aplicados ou altamente teóricos, dessa forma são apropriados para pesquisas aplicadas ou construção teórica. Para os autores existem três funções principais na criação de modelos: descrição, explicação e simulação.

3.2 Coleta de Dados: Métodos e Instrumento

A abordagem metodológica quantitativa dominou o cenário das investigações em ciências sociais e humanas até meados da década de 70, sustentada pelos ideais positivistas. Surgem então novas orientações filosóficas, novas técnicas, que indicam uma fundamentação qualitativa. Para muitos autores as pesquisas quantitativas e qualitativas não devem ser opostas, mas sim, devem convergir na complementaridade mútua.

Segundo Chizzotti (2005, p. 79), a abordagem do problema, na pesquisa qualitativa:

> parte do fundamento de que há uma relação dinâmica entre o mundo real e o sujeito, uma interdependência viva entre o sujeito e o objeto, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito.

Tanto a interpretação quanto a atribuição de significados dos fenômenos são a base no processo de pesquisa qualitativa. Flick (2004) afirma que os aspectos fundamentais da pesquisa qualitativa incidem na escolha adequada de métodos e teorias (devido a variedade existente), no reconhecimento e análise de diferentes perspectivas, nas reflexões dos pesquisadores a respeito de sua pesquisa como parte do processo de geração do conhecimento. Chizzotti (2005) afirma que muitas pesquisas qualitativas utilizam a coleta de dados quantitativos, principalmente na etapa exploratória de campo ou nas etapas em que estes dados podem mostrar uma maior relação entre fenômenos particulares.

3.2.1 Tipos de dados

Segundo Cooper e Schindler (2003, p. 83) “a coleta de dados pode variar desde uma simples observação até um levantamento grandioso em corporações multinacionais localizadas em diferentes partes do mundo”.

Os dados coletados neste trabalho são de dois tipos: primários e secundários.

- Primários: são dados brutos ou trabalhos originais, sem interpretação anterior.
- Secundários: são interpretações dos dados primários, têm pelo menos um nível de interpretação entre o fato e seu registro. São consideradas fontes de dados secundários: notícias, trabalhos como teses e dissertações, artigos científicos, entre outros documentos.

3.2.2 Técnicas de coleta e análise de dados

A coleta de dados é a etapa da pesquisa que exige um grande volume de tempo e trabalho para se reunir as informações necessárias. Pressupõe a organização cuidadosa da técnica e a elaboração de instrumentos adequados (CHIZZOTTI, 2005). Neste trabalho a coleta de dados foi realizada na forma de pesquisa bibliográfica.

A pesquisa bibliográfica ajuda a explicar o problema de pesquisa a partir de referências já publicadas. Cooper e Schindler (2003, p. 222) apresentam cinco etapas para a busca em literatura.

1. Definir o problema ou a questão de pesquisa.
2. Consultar dicionários, manuais e livros para identificar termos, pessoas ou fatos importantes para o problema.
3. Usar esses termos, pessoas ou fatos importantes para realizar buscar em índices, bibliografias e na Internet para identificar possíveis fontes secundárias específicas.

4. Localizar e rever as fontes secundárias específicas.
5. Avaliar o valor de cada fonte e de seu conteúdo.

Com base na coleta de dados das pesquisas bibliográfica, foi desenvolvido um modelo de sequenciamento que atenda aos objetivos da pesquisa. O modelo em questão terá características explicativas e de simulação. Uma vez que intenciona estender a aplicação de uma teoria bem desenvolvida, no caso algoritmo de corte de Gilmore e Gomory (1961; 1963) e de simulação por meio do esclarecimento das relações desse conceito considerando-se os tempos de *setup* dependente e as relações de atendimento de demanda no curto prazo.

3.3 Desenvolvimento do modelo

Foi desenvolvido um modelo que possibilitou sequenciar um sistema *job shop* de forma a produzir o máximo possível (de acordo com a demanda) dentro de um horizonte de planejamento de curto prazo respeitando a relação de dependência entre *setups*.

A Figura 20 apresenta de forma simplificada e ampla as etapas que compõem o processo de solução de problemas.

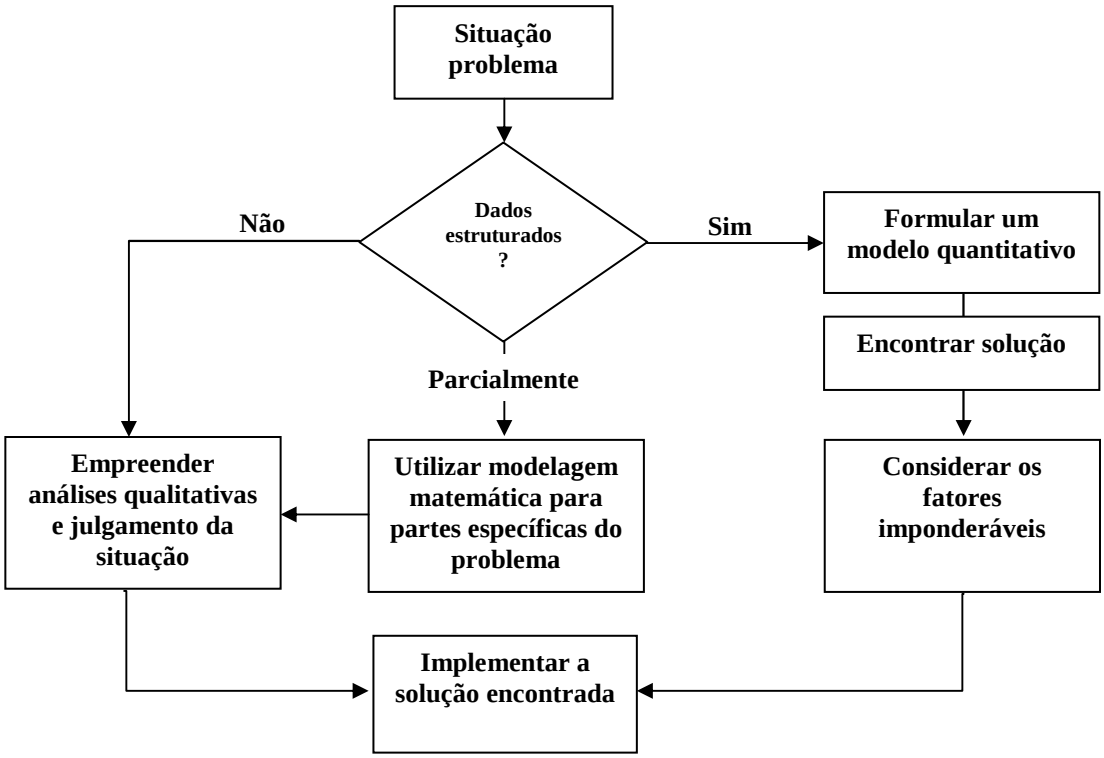

Figura 20 - Processo de solução do problema
Fonte: adaptado de Moreira (2010)

- **Definição da situação-problema**: o problema consiste em estabelecer o sequenciamento da produção em um ambiente *job shop*, levando em consideração o produto, que acarreta tempo de *setup* (dependente da sequência), ou seja, da ordem na qual os produtos são produzidos. Um subconjunto de linhas, em geral, é capaz de produzir determinado produto (produtos podem ser produzidos por linhas dispostas em paralelo). Considera-se como a demanda pelo produto é planejada em um curto horizonte de tempo (medido em horas). A Figura 21 apresenta uma visão geral do processo de produção.

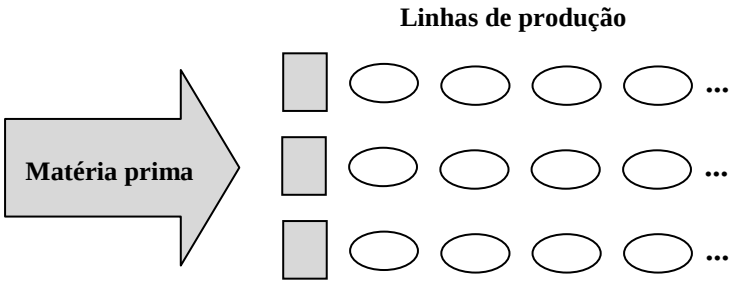

Figura 21 - Processo de produção
Fonte: elaborado pela autora

A proposta é trabalhar com um recorte do problema maior, ou seja, já se conhece quais linhas fazem quais produtos. Deve-se definir em relação as linhas de produção:

1. A quantidade de produto a ser produzido em cada linha.
2. A sequência em que os produtos são produzidos pelas linhas, minimizando os tempos de *setup*.
3. Em relação à matéria prima deve-se cuidar de forma que as quantidades existentes sejam suficientes para atender a produção das linhas.
4. As linhas têm igual velocidade de processamento.

- **Formulação de um modelo quantitativo:** o problema de corte consiste em cortar objetos disponíveis para a produção de itens de modo a atender uma demanda especificada, pode ser modelado com um problema de programação linear inteira, de grande porte, definindo os possíveis padrões de corte (POLDI; ARENALES, 2006). O problema em questão baseou-se nos trabalhos desenvolvidos por Ralph E. Gomory, no final da década de 1950 e início da década de 1960 (1958, 1960, 1961 e 1963), que propõe uma técnica de geração de colunas para a obtenção de uma solução ótima contínua. Nesse caso o padrão de corte é equivalente ao tempo de produção dos lotes(l), que são sequenciados tendo como limite superior o horizonte de planejamento

de curto prazo (L), considerando o tempo de *setup* dependente relacionado aos produtos.

- **Implementação da solução:** os experimentos computacionais foram realizados utilizando-se a ferramenta MATLAB (*MATrix LABoratory* – Laboratório de matrizes) que trabalha com programação matemática para a programação linear, programação inteira mista e programação quadrática. O APÊNDICE 01 apresenta as rotinas feitas em MATLAB.

3.4 Etapas da Pesquisa

No intuito de organizar, de forma consistente e precisa, respeitando as regras da pesquisa científica, este trabalho foi desenvolvido conforme a estrutura apresentada na Figura 22.

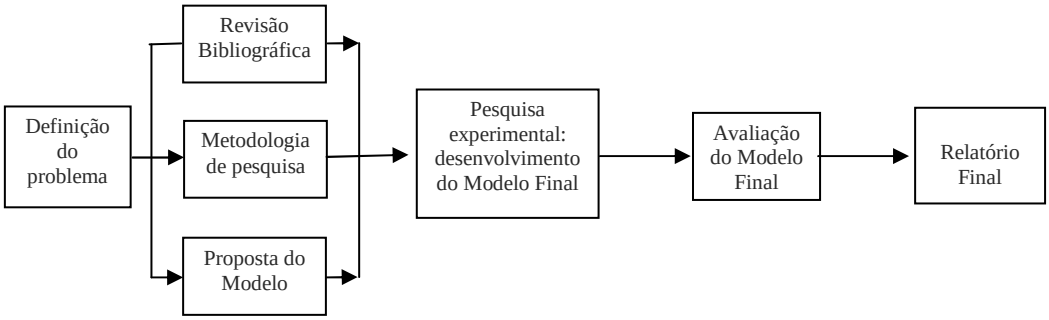

Figura 22 - Etapas da pesquisa
Fonte: elaborado pela autora

Esse trabalho iniciou-se com a definição do problema de pesquisa, Gil (1999) diz que toda pesquisa tem início com um problema, que é definido como sendo qualquer questão não resolvida e que é objeto de discussão, em qualquer domínio do conhecimento. O passo seguinte foi a revisão bibliográfica, contudo esse estudo continuou até o final da pesquisa. De forma paralela e de tal modo complementar, ocorreu a definição da metodologia da pesquisa, ou seja, quais seriam os procedimentos de pesquisa, definição das medidas, que são etapas importantes para o processo de planejamento e coleta de dados. A proposta do modelo ocorreu com base nas definições obtidas na literatura.

Na etapa seguinte, com base nesse conjunto de dados e informações desenvolveu-se o modelo final e sua avaliação. O relatório final expõe uma contribuição importante ao conhecimento e à prática de pesquisa, deixando claro que todas as evidências relevantes

foram abordadas e dessa forma dão sustentação às proposições que parametrizaram toda a investigação. O capítulo 4 apresenta o modelo matemático do problema em questão.

4 FORMULAÇÃO DOPROBLEMA

Os modelos matemáticos ou simbólicos representam uma situação real transformada em função matemática, ou seja, com dados quantitativos. São os mais utilizados para situações gerenciais. As grandezas são representadas por variáveis de decisão, e suas relações por expressões matemáticas.

De maneira geral, os problemas de programação matemática podem ser representados da seguinte forma (LACHTERMACHER, 2009).

Otimizar: $Z = f(x_1, x_2, \dots, x_n)$

$$\textbf{sujeito a} = \begin{cases} g_1(x_1, x_2, \dots, x_n) \\ g_2(x_1, x_2, \dots, x_n) \\ \vdots \\ g_m(x_1, x_2, \dots, x_n) \end{cases} \begin{cases} \leq \\ = \\ \geq \end{cases} \begin{cases} b_1 \\ b_2 \\ \vdots \\ b_m \end{cases}$$

onde:

x_j → representa a quantidade das variáveis utilizadas: (j = 1, 2, ..., n)

b_i → representa a quantidade disponível de determinado recurso: (i = 1, 2, ..., m)

X → vetor de (x_1, x_2, ..., x_n)

f(x) → função objetivo

$g_i(x)$ → funções utilizadas nas restrições do problema: (i = 1, 2, ..., m)

N → número de variáveis de decisão

M → número de restrições do modelo

Problemas de programação linear (PL) representam segundo Ragsdale (2009, p. 40) “uma categoria especial de problemas matemáticos, em que a função objetivo e todas as restrições podem ser expressas como combinações lineares das variáveis de decisão”.

Sua meta é maximizar ou minimizar uma função objetivo, respeitando sempre, as restrições indicadas. Entende-se por solução qualquer valor encontrado, dentro do domínio da função objetivo, para as variáveis de decisão. Podem existir soluções viáveis e ótimas (LACHTERMACHER, 2009).

- Solução viável: quando todas as restrições são satisfeitas.

- Solução ótima: uma solução viável que tem o valor mais favorável da função objetivo, ou seja, maximiza ou minimiza **f(x)**, podendo ser única ou não.

Algumas situações diferentes podem ser encontradas nesse tipo de problema de otimização, elas incluem múltiplas soluções ótimas, restrições redundantes, soluções ilimitadas e inviabilidade. Quando uma ou todas as variáveis de decisão em um problema de PL são restritas a assumir apenas valores inteiros, o problema resultante é chamado de problema de programação linear inteira (PLI). Muitos problemas práticos de negócios necessitam de soluções inteiras (RAGSDALE, 2009; LACHTERMACHER, 2009).

Os métodos de solução para problemas de programação inteira têm sido pesquisados por anos. Muitas abordagens foram desenvolvidas: enumeração, algoritmos de corte, e técnicas de ramificação (*branching*), são alguns dos métodos usados para resolver problemas de programação inteira.

Dados os tempos dos lotes de produtos a fabricar, suas respectivas demandas, o horizonte de planejamento e a matriz de *setup* dependente (não simétrica), trata-se de gerar a sequência de produção dos lotes (produtos) para cada linha de forma a minimizar os tempos de *setup* respeitando o horizonte de planejamento.

Para a apresentação do modelo proposto, a seguinte notação será utilizada:

L = horizonte de planejamento (medido em horas);

l_i = tempo de produção de cada lote (medido em horas);

d_i = demanda de cada lote ***i***;

a_{ij} = representa o número de vezes que o item ***i*** será cortado no tamanho l_i;

s_{uv} = *setup* dependente se o lote ***u*** for seguido em sua produção pelo lote ***v.***

Considerando um conjunto de **n** padrões de corte (tempo de produção dos lotes) previamente definidos, formula-se o modelo do problema de corte unidimensional como um problema de programação linear inteira com o objetivo de minimizar o tempo ocioso (não produtivo).

Minimizar

$$\sum_{j=1}^{n} x_j$$

sujeito a

$$\sum_{j=1}^{n} a_{ij}\ x_j = d_i, \quad i = 1,..,m$$

$$x_j \geq 0 \text{ e inteiro}, j = 1,..,n$$

Cada variável $\mathbf{x_j}$ representa a quantidade do padrão de corte do tipo **j** a ser usado no planejamento da produção. Verifica-se que esse problema (com restrições de igualdade) é equivalente ao problema de minimizar o desperdício.

O desperdício (tempo não usado da linha) com o padrão de corte **j** é dado pelo somatório do padrão de corte (l_i) multiplicado pela quantidade de vezes que foi cortado (a_{ij}) subtraindo-se do horizonte de planejamento (L).

$$L - \sum_{i=1}^{m} a_{ij} l_i$$

O **desperdício total** obtido com todos os padrões é dado por:

$$L \sum_{j=1}^{n} x_j - \sum_{i=1}^{m} l_i \, d_i$$

Como **L, l, d** são constantes, o problema de minimizar o desperdício é equivalente ao de maximizar o número de objetos processados.

Na formulação do problema de corte unidimensional, a função objetivo pode incorporar o tempo de *setup*. Este termo se refere ao tempo da troca de padrão, isto é, a cada padrão novo a ser utilizado é necessário fazer um ajuste e reposicionamento dos equipamentos, o que exige tempo e trabalho.Incorporando o tempo de *setup* ao problema tem-se o modelo:

Minimizar

$$\sum_{j=1}^{n} \boldsymbol{x_j} + \sum_{j=1}^{n} \boldsymbol{s_{uv}(x_j)}$$

sujeito a

$$\sum_{j=1}^{n} a_{ij} \; x_j = d_i, \quad i = 1, .., m$$

$$x_j \geq 0 \text{ e inteiro}, j = 1, .., n$$

Fazendo a união do número de objetos processados, desperdício e *setup* na função objetivo tem-se um problema de minimização como abaixo:

Minimizar

$$L\sum_{j=1}^{n} x_j - \sum_{i=1}^{m} l_i\, d_i + \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{s_{uv}(x_j)}$$

sujeito a

$$\sum_{j=1}^{n} a_{ij}\; x_j = d_i, \quad i = 1,..,m$$

$$x_j \geq 0 \text{ e inteiro}, j = 1,..,n$$

Para formular o modelo matemático, do problema proposto, foram consideradas simplificações do sistema, com isso pode-se afirmar que o modelo matemático é uma representação simplificada (abstração) do problema real. Ele é suficientemente detalhado para deter os elementos essenciais do problema, mas suficientemente tratável por método de resolução (ARENALES *et al.*, 2007).

Foram aplicadas técnicas matemáticas e tecnologia, de forma a resolver o problema matemático e então visualizar quais considerações ele sugere. No próximo capítulo será descrito como foi implementado o modelo aqui apresentado.

5 MÉTODO DE RESOLUÇÃO

Pode-se afirmar, com base na teoria apresentada no capítulo 2, que o modelo proposto nesta tese gera uma solução ótima de enumeração implícita e diferencia-se pelas particularidades de sua implementação e eficiência computacional. Desenvolveu-se um algoritmo com base em Gilmore e Gomory (1961; 1963) e seu método de geração de colunas para minimizar os tempos ociosos (custos) em ambientes *job shop* com tempos de *setup* dependentes.

Se valida a escolha de utilizar o método de Gilmore e Gomory (1961; 1963) pelo fato de que este método possibilita trabalhar apenas com o subconjunto de padrões que tem uma propriedade em comum: todas as variáveis associadas às colunas geradas são candidatas a estar na base da solução, ou seja, ao invés de se trabalhar com todo o universo de colunas, trabalha-se somente com as melhores opções (DAL GALLO, 2008).

Implementou-se, usando MATLAB (versão educacional – R2010a), de acordo com os passos indicados abaixo o algoritmo de resolução do problema proposto:

<u>Passo 1</u>: Determinar a matriz básica inicial da seguinte forma: A = diagonal (L/l_i)

- Gerar os cortes homogêneos para a primeira iteração - um padrão de corte homogêneo é o padrão que produz apenas um tipo de item.

$$A = (a_1, ..., a_i) = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & a_{ii} \end{bmatrix}$$

<u>Passo 2</u>: Determinar uma partição básica factível **A = [BN]**.

- Criar os vetores de índices básicos e não-básicos: **(B_1, B_2, B_3)** e **(N_1, N_2, N_3)**.
- Fazer Iteração = 1.

<u>Passo 3</u>: Calcular a solução básica (x_B)

- Solução geral = $x_B = (B^{-1}b) - (B^{-1}Nx_N)$
- Uma vez que se atribui zeros para as variáveis fixadas $(B^{-1}Nx_N) = 0$, então

$$\boldsymbol{x_B = (B^{-1}b)}$$

  Onde, B^{-1}é a inversa de B e b é o vetor de demandas.
- A Solução é factível se as variáveis respeitarem a condição de não negatividade, caso contrário indicar solução não factível.

<u>Passo 4</u>: Calcular o vetor multiplicador simplex (λ)

$$\lambda^T = \boldsymbol{c_B^T B^{-1}}$$

<u>Passo 5</u>: Resolver o problema da mochila e obter o novo padrão de corte.

- Na mochila oλ faz o papel dos valores de utilidade dos itens.
- Com base no método das economias de Clarke e Wright (1963 *apud* BALLOU, 2006), serão gerados os arranjos possíveis dos produtos (l_i):

$$\boldsymbol{A_{n,p} = \frac{n!}{(n-p)!}}$$

  Onde, n é a quantidade de produtos e p = 2 {sempre serão gerados aos pares}.

O método das economias de Clarke e Wright (1963) foi desenvolvido pensando no problema de roteamento de veículos e caracteriza-se por considerar que a partir de um depósito central deverão sair todos os produtos a serem entregues aos clientes nas respectivas quantidades. Está disponível para tal, um número específico de veículos, cada um com certa capacidade. Todo veículo que é usado na solução deverá realizar uma rota, começando e terminando no depósito e as entregas poderão ser feitas a um ou mais clientes. O objetivo desse método e encontrar uma solução que minimize os custos de transporte total. O conceito de economia é representado na Figura 23, por meio da união de duas rotas em uma rota, isso ocorre mediante a consolidação das paradas em uma rota (BALLOU, 2006). Dada a necessidade de ordenar os *setups* dependentes para minimizar os desperdícios foi usada a similaridade com o problema das economias de Clarke e Wright (1963).

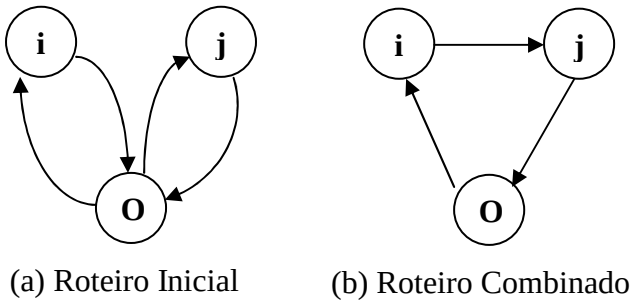

Figura 23 - Conceito de economia
Fonte: Adaptado de Ballou (2006).

Na Figura 23(a) os clientes **i** e **j** são visitados em rotas separadas e na Figura 23(b) as rotas estão combinadas em um único roteiro.

Com base nesse modelo foi gerada uma rotina de ordenação dos produtos (l_i) aos pares:

- São gerados $\boldsymbol{A_{n,p}}$arranjos possíveis;
- Para cada par é verificado e atribuído o valor do *setup* dependente;
- Os pares são ordenados do menor para o maior valor de *setup*;
- Cada rota é criada considerando-se primeiro o par de menor valor de *setup*, o segundo valor do primeiro par indica o primeiro valor do próximo par a ser selecionado e assim sucessivamente, até que todos os produtos (l_i) sejam considerados na rota. Essa rotina se repete até que todos os pares sejam selecionados.
- Não é permitido o fechamento da rota durante a seleção dos pares, ou em sua finalização.

- Após a geração das rotas ou sequências, é feita uma ordenação do menor para o maior valor de *setup* (somatório dos valores a partir do pares que compõem a rota).
- A cada iteração, é selecionada uma única vez a sequência de menor valor de *setup*. Nas iterações seguintes serão selecionadas as próximas sequências de menor valor, e assim sucessivamente.
- Com a sequência selecionada é feita uma ordenação dos dados da mochila e então somados os tempos de *setup* dependente correspondentes.
- É gerado um novo padrão de corte.

<u>Passo 6</u>: Calcular o custo relativo

$$\hat{c}_{Nj} = c_{Nj} - \lambda^T a_{Nj} \qquad j = 1,2,\dots,n-m$$

- Se $\hat{c}_{Nj} \geq 0$, então **pare** {solução na iteração atual é ótima}

<u>Passo 7</u>: Calcular direção simplex

y = $B^{-1}a_k$

Onde, k = índices das variáveis não básicas.

<u>Passo 8</u>: Determinar o tamanho do passo e variável a sair da base.

- Se y (direção simplex) ≤ 0, então **pare** {problema não tem solução ótima finita: $f(x) \rightarrow -\infty$}.
- Caso contrário, determine a variável a sair da base pela razão mínima: $\hat{\varepsilon}$ = min $\left\{\frac{\hat{x}_{B_1}}{y_1}, \frac{\hat{x}_{B_2}}{y_2}, \frac{\hat{x}_{B_3}}{y_3}\right\}$

<u>Passo 9</u>: Atualização – nova partição básica.

- Nova matriz B.
- Iteração = Iteração + 1.
- Voltar ao passo 2.

Resumindo, o problema proposto foi resolvido usando-se como base o algoritmo de Gilmore e Gomory (1961; 1963), com a criação de um novo padrão de geração de colunas.

Esse novo padrão foi desenvolvido a partir de uma heurística do método das economias de Clarke e Wright (1963) e a nova coluna gerada considera os tempos de *setup* dependente relacionado aos produtos. Essa abordagem é importante uma vez que os sistemas de produção *job shop* para cada lote de um diferente produto têm suas máquinas e ferramentas ajustadas, adaptadas e arranjadas, além do fato de que em sistemas com essas características os custos com *setup* tendem a serem maiores (FERNANDES; GODINHO FILHO, 2010).

No capítulo seguinte serão apresentados os resultados computacionais e suas interpretações.

6 RESULTADOS E DISCUSSÃO

Considerando-se que uma das quatro principais atividades do controle da produção é programar e sequenciar as tarefas nas máquinas. Intencionou-se por meio do algoritmo desenvolvido otimizar a programação das linhas, uma vez que, a produção em ambiente *job shop* impõe a necessidade de um plano de produção bem feito e que possa integrar novos lotes de produção à medida que outros sejam completados, ou seja, nesse ambiente produtivo o plano de produção deve ser constantemente replanejado e atualizado, pois o sucesso do processo produtivo depende diretamente do plano de produção (FERNANDES; GODINHO FILHO, 2010).

Nos experimentos analisados, os dados referentes a demanda dos produtos e ao tempo de produção de cada produto(l_i) foram extraídos de exemplos da literatura (BAKER, 1974; CONWAY; MAXWELL; MILLER, 2003; PINEDO, 2008). Considerou-se o L=24 e a matriz não simétrica dos tempos de *setup* dependente foi gerada aleatoriamente para cada simulação. Foi usada a sigla GILGOCW, para identificar o algoritmo desenvolvido. A intenção é que por meio dos experimentos possa-se relatar as principais características das soluções geradas pelo modelo proposto. Os parâmetros para gerar um problema são os seguintes:

L = horizonte de planejamento (medido em horas);

l_i = tempo de produção de cada produto (medido em horas);

d_i = demanda de cada lote (medida por item);

ST = Matriz de *setup* dependente (fração de hora).

No **primeiro problema** usaram-se os dados apresentados no Quadro 11, sendo que a primeira linha é composta dos tempos de produção de um lote de cada produto e a segunda linha pelas respectivas quantidades demandadas. O horizonte de planejamento (L) foi considerado igual a 24 horas. Foram considerados para esse problema 5 produtos.

Quadro11 - Dados do problema 1: tempos de produção (l_i) e respectivas demandas (d_i)

	P1	**P2**	**P3**	**P4**	**P5**
l_i	2	3	4	6	9
d_i	26	8	12	4	2

Fonte: Elaborado pela autora.

No Quadro 12 mostra-se a matriz com os tempos de *setup* dependente dos cinco produtos, que foram gerados aleatoriamente entre 0 e 1 (indicando a fração de hora correspondente).

Quadro 12 - Os tempos de *setup* dependente do problema 1

	P1	**P2**	**P3**	**P4**	**P5**
P1	**0**	0,0747	0,4844	0,4902	0,4730
P2	0,1095	**0**	0,2159	0,0451	0,0393
P3	0,3767	0,5000	**0**	0,4866	0,4788
P4	0,3381	0,1529	0,1571	**0**	0,1220
P5	0,2857	0,0271	0,4709	0,4037	**0**

Fonte: Elaborado pela autora.

Os cortes gerados pelo GILGOCW indicam as programações de cada linha:

- A**linha 1** deve produzir **12 lotes** do **produto 1** em **24 horas**, ou seja, dentro do horizonte de planejamento. Com isso a perda é igual a zero.
- A **linha 2** deve produzir **8 lotes** do **produto 2** em **24 horas.** Com isso a perda é igual a zero.
- A **linha 3** deve produzir **6 lotes** do **produto 3** em **24 horas**. Com isso a perda é igual a zero.
- A **linha 4** deve produzir **4 lotes** do **produto 4** em **24 horas**. Com isso a perda é igual a zero.
- A **linha 5** deve produzir **2 lotes** do **produto 1** e na sequência deve produzir **2 lotes** do **produto 5**. São **22 horas** de produção com 2 horas de tempo ocioso.

A programação das linhas foi gerada com o objetivo de minimizar o desperdício dos tempos não produtivos (*setup* e folga). Os cortes não homogêneos indicam que houve o aproveitamento das linhas de forma inteligente, distribuindo o tempo de forma a produzir mais de um produto minimizando o tempo ocioso. O valor do cálculo do percentual do tempo ocioso resultante da programação indicada é de 1,67%, foram perdidas duas horas em 120

horas, considerando-se o somatório dos tempos das cinco linhas. Em relação a linha 5 o percentual de perda é de 8,33%.

O Gráfico 1 possibilita a visualização dos percentuais de ocupação de cada linha, de quais produtos estão sendo produzidos e em que sequência estão sendo produzidos. Também permite a percepção do tempo ocioso indicado na linha 5.

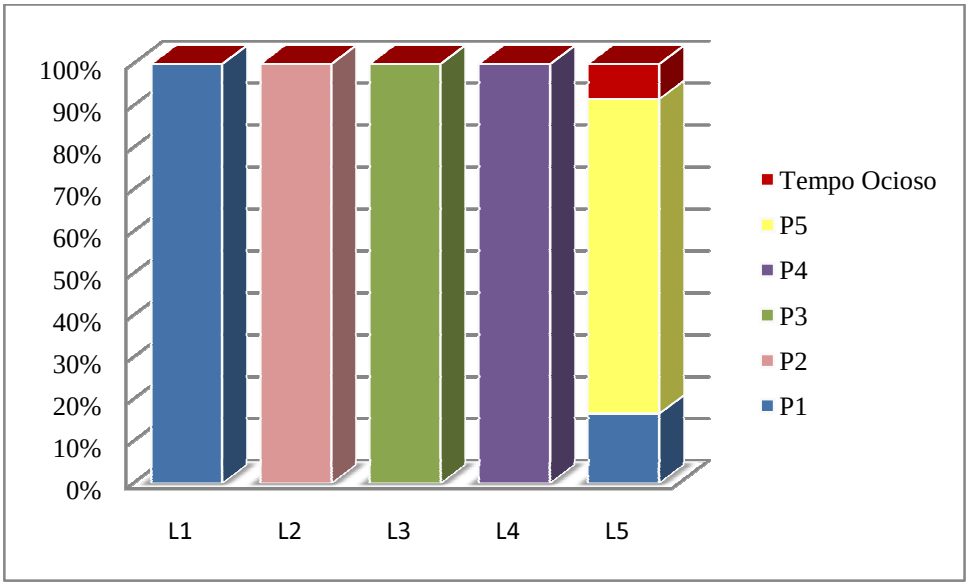

Gráfico 1 - Problema 1: programação de cada linha
Fonte: Elaborado pela autora.

De forma a atender a demanda, cada linha deve repetir a programação indicada um determinado número de vezes: A **linha 1** deve repetir a programação **2 vezes**; a **linha 2** deve repetir a programação **1 vez**; a **linha 3** deve repetir a programação **2 vezes**; a **linha 4** deve repetir a programação **1 vez** e a **linha 5** deve repetir a programação **1 vez.**

O Quadro 13 mostra os dados da solução gerada.

Quadro 13 - Problema 1: dados da solução gerada

Linhas	**L1**	**L2**	**L3**	**L4**	**L5**	**Demanda**
x_B	2	1	2	1	1	
P1	**12**	0	0	0	**2**	26
P2	0	**8**	0	0	0	8
P3	0	0	**6**	0	0	12
P4	0	0	0	**4**	0	4
P5	0	0	0	0	**2**	2
Tempo não produtivo	0	0	0	0	**2**	

Fonte: Elaborado pela autora.

Tem-se nele os dados referentes à programação de cada linha (apresentado nas colunas L1, ..., L5) e cada produto (apresentado nas linhas P1, ..., P5). O cálculo do **tempo ocioso** é gerado pela subtração do tempo de produção do tempo de horizonte (L) de cada linha. A variável $\boldsymbol{x_B}$ indica o número de vezes que a programação deverá ser repetida de forma a atender a demanda. Pode-se notar nessa solução que foi feita a distribuição e o arranjo dos produtos entre as linhas de forma a minimizar o tempo ocioso. O algoritmo GILGOCW gerou a solução apresentada em 0.010133 segundos (processador *Intel Core Duo*) com duas iterações.

Cabe observar que foram obtidas outras três soluções usando os dados apresentados no Quadro 11, com a geração de novos tempos de *setup* de forma aleatória. Elas geraram um maior desperdício de tempo, respectivamente: 2,5% (3 horas de tempo ocioso na L5); 3,33% (4 horas de tempo ocioso na L5) e 5% (6 horas de tempo ocioso na L5 e a distribuição das linhas foi feita com cortes homogêneos).

No **segundo problema** usaram-se os dados apresentados no Quadro 14, sendo que a primeira linha é composta dos tempos de produção de um lote de cada produto e a segunda linha pelas respectivas quantidades demandadas. O horizonte de planejamento (L) foi considerado igual a 24 horas. Foram considerados para esse problema 6 produtos.

Quadro 14 - Dados do problema 2: tempos de produção (l_i) e respectivas demandas (d_i)

	P1	**P2**	**P3**	**P4**	**P5**	**P6**
l_i	6	4	8	10	12	2
d_i	8	24	18	8	12	4

Fonte: Elaborado pela autora.

O Quadro 15 mostra a matriz com os tempos de *setup* dependente dos seis produtos, que foram gerados aleatoriamente entre 0 e 1 (indicando a fração de hora correspondente).

Quadro 15 - Os tempos de *setup* dependente do problema 2

	P1	**P2**	**P3**	**P4**	**P5**	**P6**
P1	**0**	0,1605	0,1534	0,1810	0,0901	0,1491
P2	0,0205	**0**	0,0138	0,3188	0,0945	0,2046
P3	0,2033	0,3001	**0**	0,2334	0,1797	0,3334
P4	0,1105	0,2507	0,2697	**0**	0,3770	0,0962
P5	0,2804	0,2382	0,2358	0,2100	**0**	0,0091
P6	0,3066	0,0289	0,4000	0,1670	0,1165	**0**

Fonte: Elaborado pela autora.

Os cortes gerados pelo GILGOCW indicam as programações de cada linha:

- A**linha 1**deve produzir **4 lotes** do **produto 1** em **24 horas**. Com isso a perda é igual a zero.
- A **linha 2** deve produzir **6 lotes** do **produto 2** em **24 horas.** Com isso a perda é igual a zero.
- A **linha 3** deve produzir **3 lotes** do **produto 3** em **24 horas.**Com isso a perda é igual a zero.
- A **linha 4** deve produzir **2 lotes** do **produto 4** e na sequência deve produzir **1 lote** do **produto 6**. São **22 horas** de produção com 2 horas de tempo ocioso.
- A **linha 5** deve produzir **2 lotes** do **produto** em **24 horas**. Com isso a perda é igual a zero.
- A **linha 6 não foi utilizada.**

A programação gerada para o segundo problema otimizou a utilização das linhas por meio da não utilização da linha 6 e também minimizou a perda dos tempos não produtivos (*setup* e folga). O valor do cálculo do percentual do tempo ocioso resultante da programação indicada é de 1,39%, foram perdidas duas horas em 144 horas, considerando-se o somatório dos tempos das seis linhas. Em relação a linha 4o percentual de perda é de 8,33%.

O Gráfico 2 possibilita a visualização dos percentuais de ocupação de cada linha, de quais produtos estão sendo produzidos e em que sequência estão sendo produzidos. Também permite a percepção do tempo ocioso indicado na linha 4 e a não utilização da linha 6.

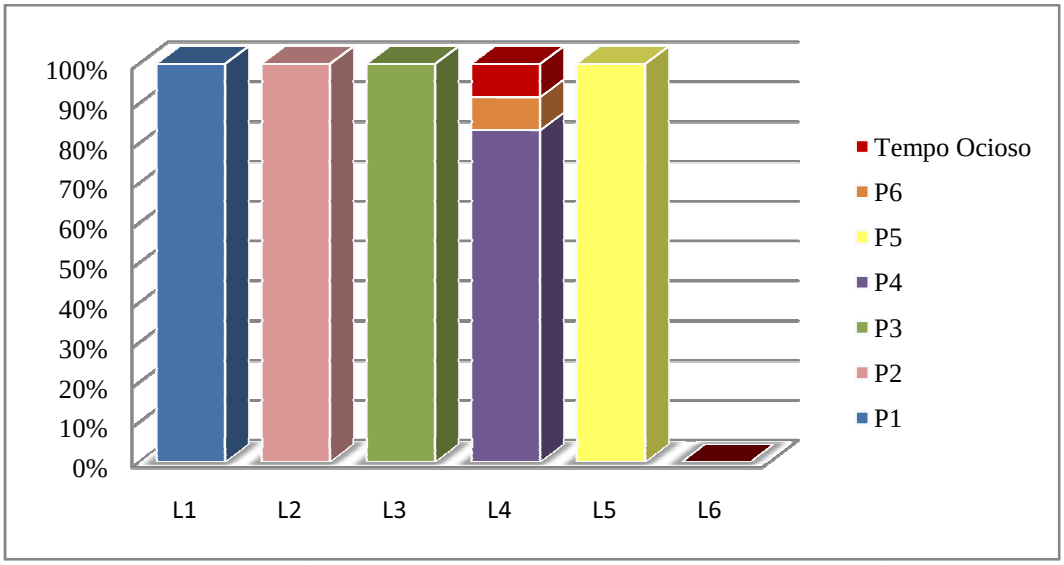

Gráfico 2 - Problema 2: programação de cada linha
Fonte: Elaborado pela autora.

De forma a atender a demanda indicada, foram utilizadas 5 das 6 linhas e cada uma deve repetir a programação indicada um determinado número de vezes: A **linha 1**deve repetir a programação **2 vezes**; a **linha 2** deve repetir a programação **4 vezes**; a **linha 3** deve repetir a programação **6 vezes**; a **linha 4** deve repetir a programação **4 vezes**; a **linha 5** deve repetir a programação **6 vezes** e a **linha 6 não foi utilizada.**

O Quadro 16 mostra os dados da solução indicada.

Quadro 16 - Problema 2: dados da solução gerada

Linhas	**L1**	**L2**	**L3**	**L4**	**L5**	**L6**	**Demanda**
x_B	2	4	6	4	6	**0**	
P1	**4**	0	0	0	0	**0**	8
P2	0	**6**	0	0	0	**0**	24
P3	0	0	**3**	0	0	**0**	18
P4	0	0	0	**2**	0	**0**	8
P5	0	0	0	0	**2**	**0**	12
P6	0	0	0	**1**	0	**0**	4
Tempo não produtivo	0	0	0	**2**	0	0	

Fonte: Elaborado pela autora.

O algoritmo GILGOCW gerou a solução apresentada em 0.107497 segundos (processador *Intel Core Duo*) com duas iterações.

Cabe observar que foi obtida outra solução usando os dados apresentados no Quadro 14, com a geração de novos tempos de *setup* de forma aleatória. Ela gerou um maior desperdício de tempo: 2,78% com 4 horas de tempo ocioso na L4, fazendo a distribuição de todas as linhas com cortes homogêneos.

No **terceiro problema** usaram-se os dados apresentados no Quadro 17, sendo que a primeira linha é composta dos tempos de produção de um lote de cada produto e a segunda linha pelas respectivas quantidades demandadas. O horizonte de planejamento (L) foi mantido igual a 24 horas. Foram considerados para esse problema 10 produtos.

Quadro 17 - Dados do problema 3: tempos de produção (l_i) e respectivas demandas (d_i)

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**
l_i	1	2	4	6	8	12	24	5	10	7
d_i	24	4	24	8	18	12	2	16	8	6

Fonte: Elaborado pela autora.

O Quadro 18 mostra a matriz com os tempos de *setup* dependente dos dez produtos, que foram gerados aleatoriamente entre 0 e 1 (indicando a fração de hora correspondente).

Quadro 18 - Os tempos de *setup* dependente do problema 3

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**
P1	**0**	0,2162	0,5705	0,3832	0,5119	0,1357	0,5897	0,2688	0,5623	0,5863
P2	0,2270	**0**	0,0296	0,5817	0,0233	0,3380	0,3190	0,2997	0,6000	0,5141
P3	0,0702	0,2093	**0**	0,4595	0,0585	0,2770	0,0687	0,0180	0,5935	0,4583
P4	0,3050	0,2780	0,3848	**0**	0,5769	0,3713	0,3957	0,3484	0,1854	0,4278
P5	0,4971	0,1449	0,0442	0,5173	**0**	0,3310	0,1162	0,3483	0,4586	0,1501
P6	0,2090	0,4692	0,1289	0,4965	0,3239	**0**	0,5214	0,4513	0,1256	0,0243
P7	0,3034	0,1664	0,2190	0,0804	0,5499	0,1741	**0**	0,0466	0,5241	0,2812
P8	0,0718	0,3195	0,2596	0,4629	0,2069	0,5349	0,0629	**0**	0,4003	0,0714
P9	0,3502	0,3232	0,4613	0,3756	0,4040	0,5699	0,1804	0,1177	**0**	0,0983
P10	0,3137	0,5314	0,5572	0,2420	0,4426	0,4906	0,4862	0,3231	0,3532	**0**

Fonte: Elaborado pela autora.

Os cortes gerados pelo GILGOCW indicam as programações de cada linha:

- A**linha 1** deve produzir **24 lotes** do **produto 1** em **24 horas**. Com isso a perda é igual a zero.
- A **linha 2** deve produzir **1 lote** do **produto 2** e na sequência deve produzir **2 lotes** do **produto 9**. São **22 horas** de produção com 2 horas de tempo ocioso.
- A **linha 3** deve produzir **6 lotes** do **produto 3** em **24 horas**. Com isso a perda é igual a zero.
- A **linha 4** deve produzir **4 lotes** do **produto.** Com isso a perda é igual a zero.
- A **linha 5** deve produzir **3 lotes** do **produto 5** em **24 horas**. Com isso a perda é igual a zero.
- A **linha 6** deve produzir **2 lotes** do **produto 6**.Com isso a perda é igual a zero.
- A **linha 7**deve produzir **1 lote** do **produto 7**.Com isso a perda é igual a zero.
- A **linha 8** deve produzir **4 lotes** do **produto 8**. São **20 horas** de produção com 4 horas de tempo ocioso.
- A **linha 9 não foi utilizada.**
- A **linha 10** deve produzir **3 lotes** do **produto 10**. São **21 horas** de produção com 3 horas de tempo ocioso.

Mais uma vez a programação gerada para o terceiro problema otimizou a utilização das linhas por meio da não utilização da linha 9 e também minimizou a perda dos tempos não produtivos (*setup* e folga), por meio da geração de cortes não homogêneos. O valor do cálculo

do percentual do tempo ocioso resultante da programação indicada é de 3,75%, foram perdidas 9 horas em 240 horas, considerando-se o somatório dos tempos das dez linhas. Em relação as linhas 2, 8 e 10 o percentual de perda é respectivamente de 8,33%, 16,67% e 12,5%.

O Gráfico 3 permite a visualização da distribuição dos trabalhos entre as dez linhas, do percentual de ocupação de cada linha, de quais produtos estão sendo produzidos e em que sequência estão sendo produzidos. Também permite a percepção do tempo ocioso indicado nas linhas 2, 8 e 10 e a não utilização da linha 9.

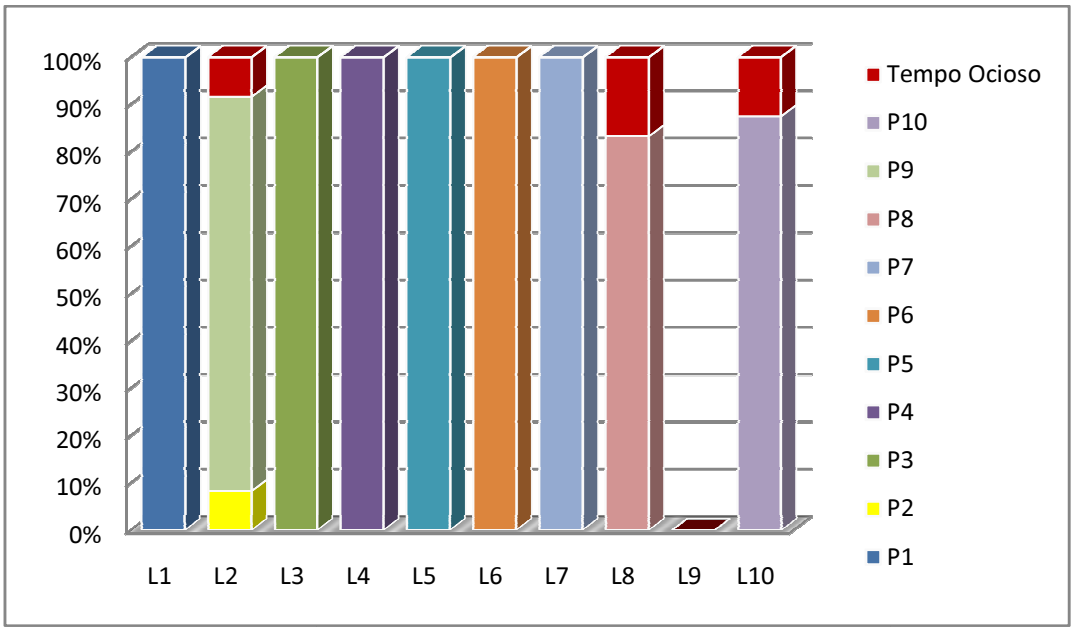

Gráfico 3 - Problema 3: programação de cada linha
Fonte: Elaborado pela autora.

De forma a atender a demanda indicada, foram utilizadas 9 das 10 linhas e cada uma deve repetir a programação indicada um determinado número de vezes: A **linha 1**deve repetir a programação **1 vez**; a **linha 2** deve repetir a programação **4 vezes**; a **linha 3** deve repetir a programação **4 vezes**; a **linha 4** deve repetir a programação **2 vezes**; a **linha 5** deve repetir a programação **6 vezes**; a **linha 6** deve repetir a programação **6 vezes**; a **linha 7** deve repetir a programação **2 vezes**; a **linha 8** deve repetir a programação **4 vezes**; a **linha 9 não foi utilizada** e a **linha 10** deve repetir a programação **2 vezes.**

O Quadro 19 mostra os dados da solução indicada. Pode-se novamente notar nessa solução que foi feita a distribuição e a organização dos produtos entre as linhas de forma a minimizar o tempo ocioso e liberação de uma das linhas.

Quadro 19 - Problema 3: dados da solução gerada

Linhas	**L1**	**L2**	**L3**	**L4**	**L5**	**L6**	**L7**	**L8**	**L9**	**L10**	**Demanda**
x_B	1	4	4	2	6	6	2	4	**0**	2	
P1	**24**	0	0	0	0	0	0	0	**0**	0	24
P2	0	**1**	0	0	0	0	0	0	**0**	0	4
P3	0	0	**6**	0	0	0	0	0	**0**	0	24
P4	0	0	0	**4**	0	0	0	0	**0**	0	8
P5	0	0	0	0	**3**	0	0	0	**0**	0	18
P6	0	0	0	0	0	**2**	0	0	**0**	0	12
P7	0	0	0	0	0	0	**1**	0	**0**	0	2
P8	0	0	0	0	0	0	0	**4**	**0**	0	16
P9	0	2	0	0	0	0	0	0	**0**	0	8
P10	0	0	0	0	0	0	0	0	**0**	**3**	6
Tempo não produtivo	0	**2**	0	0	0	0	0	**4**	**0**	**3**	

Fonte: Elaborado pela autora.

O algoritmo GILGOCW gerou a solução apresentada em 0.099231 segundos (processador *Intel Core Duo*) com duas iterações.

Cabe observar que foram obtidas outras duas soluções usando os dados apresentados no Quadro 17, com a geração de novos tempos de *setup* de forma aleatória. Elas geraram maior desperdício de tempo, respectivamente: 4,17% (o somatório, das perdas em L8, L9 e L10, totaliza 10 horas de tempo ocioso) e 4,58% (o somatório, das perdas em L8, L9 e L10, totaliza 11 horas de tempo ocioso).

Analisando os dados dos três experimentos, pode-se notar que o tempo computacional exigido pelo algoritmo para gerar a solução dos três problemas foi de 0.010133 segundos para 5 produtos, 0.107497 segundos para 6 produtos e 0.099231 segundos para 10 produtos. Isto foi possível porque o algoritmo usou uma heurística para verificação das sequências de *jobs* com os menores tempos de *setup* dependentes, ou seja, foram criadas $\frac{n!}{(n-p)!}$ sequências das quais foram selecionadas as com o menor $\sum$ dos tempo de *setup* dependente. Nos três experimentos, o seqüenciamento, com menor tempo de *setup* e tempo ocioso, foi encontrado com duas iterações.

O Quadro 20 mostra a média dos tempos computacionais para problemas com até 50 produtos.

Quadro 20 - Tempos computacionais médios

Quantidade de Produtos	Tempo médio computacional (processador *Intel Core Duo*,3 GB Memória RAM)
20 produtos	0,83 segundos
25 produtos	2,02 segundos
30 produtos	4,13 segundos
40 produtos	13,47 segundos
50 produtos	35,17 segundos

Fonte: Elaborado pela autora.

Levando-se em consideração o tempo médio computacional para problema com até 50 produtos, pode-se considerar sua aplicabilidade em problemas reais.

Como pode ser visto no Gráfico 4, a comparação entre os desperdícios gerados e o tempo total disponível (somatório dos tempos das linhas) em cada problema apresentado, permite afirmar que o modelo é eficaz em identificar soluções de sequenciamento que minimizem os tempos não produtivos.

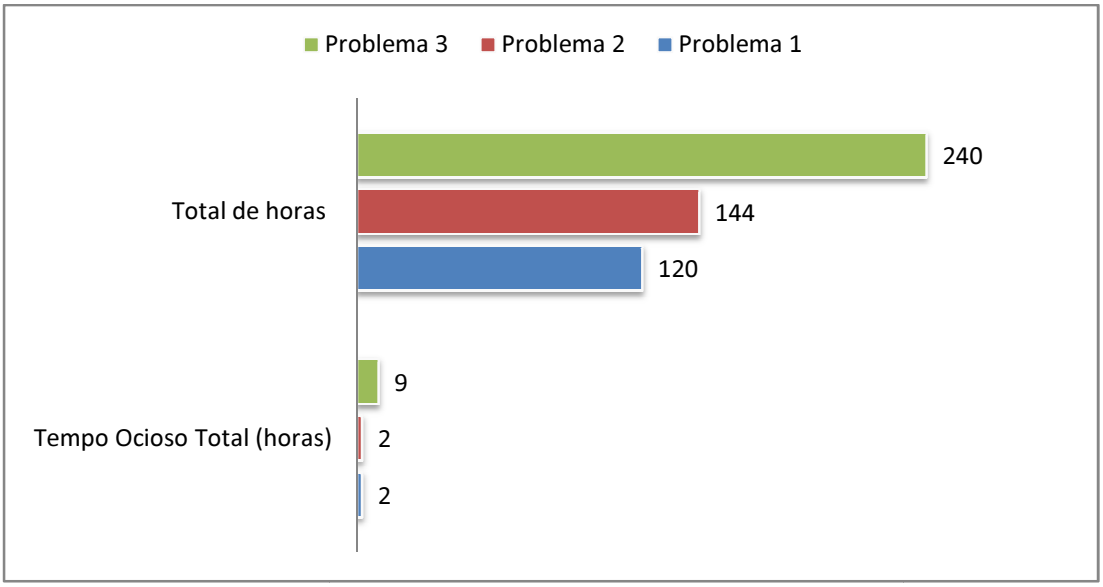

Gráfico 4 - Horas disponíveis x tempo ocioso
Fonte: Elaborado pela autora.

O *setup* ou tempo de preparação da máquina para poder iniciar-se outra tarefa, influencia na escolha dos métodos a serem utilizados nas atividades de coordenação de ordens e programação da produção. Pode-se verificar na literatura que a consideração de tempos de *setup* em problemas *job shop* começou cerca de três décadas atrás, com a identificação por

meio de simulações que tempos de *setup* dependentes possuem um papel crítico no desempenho de uma operação *job shop* (ALLAHVERDI; GUPTA; ALDOWAISAN, 1999).

Nos experimentos realizados observou-se a influencia do *setup* na geração das soluções e no aumento da perda dos tempos não produtivos. O aumento das perdas com a inserção dos tempos de *setup* dependente é apresentado no **quarto problema**.

Em um quarto problema usaram-se os dados apresentados no Quadro 21, com o horizonte de planejamento (L) igual a 5 horas, para a programação de 3 produtos e matriz de *setup* dependente zerada.

Quadro 21 - Dados do problema 4: tempos de produção (l_i) e respectivas demandas (d_i)

	P1	**P2**	**P3**
l_i	2	3	4
d_i	20	10	20

Fonte: Elaborado pela autora.

O valor do cálculo do percentual do tempo ocioso resultante da programação indicada é de 13,33%, foram perdidas duas horas em 15 horas, considerando-se o somatório dos tempos das três linhas. Em relação as linhas 1 e 3 o percentual de perda é de 20%.

O Gráfico 5 possibilita a visualização dos percentuais de ocupação de cada linha, de quais produtos estão sendo produzidos e em que sequência estão sendo produzidos. Também permite a percepção dos tempos ociosos indicados nas linhas 1 e 3.

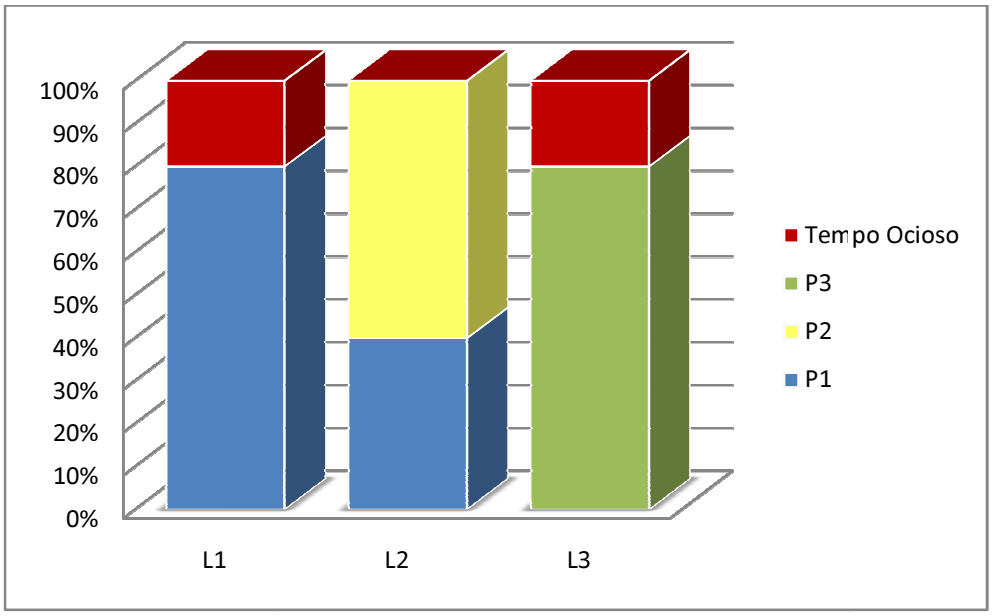

Gráfico 5 - Problema 4: programação de cada linha sem *setup*
Fonte: Elaborado pela autora.

O Quadro 22 mostra a matriz com os tempos de *setup* dependente dos três produtos, que foram gerados aleatoriamente entre 0 e 1 (indicando a fração de hora correspondente).

Quadro 22 - Os tempos de *setup* dependente do problema 4

	P1	**P2**	**P3**
P1	**0**	0,382	0,116
P2	0,378	**0**	0,228
P3	0,053	0,041	**0**

Fonte: Elaborado pela autora.

O valor do cálculo do percentual do tempo ocioso resultante da programação indicada é de 26,67%, foram perdidas quatro horas em 15 horas, considerando-se o somatório dos tempos das três linhas. Em relação as linhas 1 e 3 o percentual de perda é de 20% e linha 2 o percentual de perda é de 40%.

Acrescentando-se os tempos de *setup* dependente apresentados no Quadro 22, tem-se um aumento nos percentuais de perda de 49%, o que fortalece a afirmação de que os tempos de *setup* dependentes possuem um papel crítico no desempenho de uma operação *job shop*. O Gráfico 6 possibilita a visualização do aumento percentuais de perda nas três linhas.

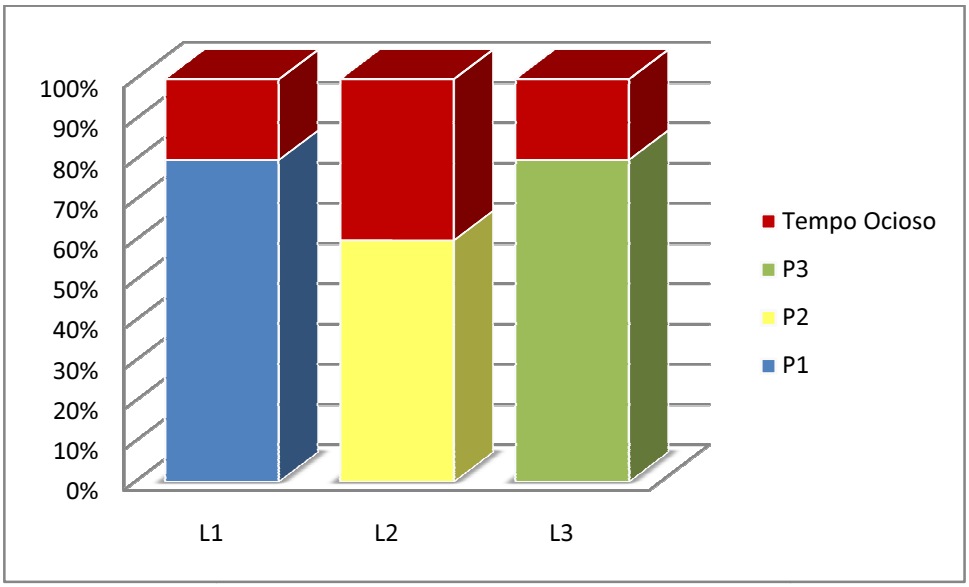

Gráfico 6 - Problema 4: programação de cada linha com *setup*
Fonte: Elaborado pela autora.

Segundo Reid e Sanders (2005) o modo como as tarefas são programadas reflete a estratégia competitiva da empresa e servem como ferramenta para manter o sincronismo de todas as áreas funcionais. Assim, pode-se afirmar que uma empresa que tem uma programação eficaz utiliza seus recursos de forma inteligente e assim alcança seus objetivos.

7 CONSIDERAÇÕES FINAIS

Esta pesquisa teve como intenção identificar e analisar um método que possibilitasse sequenciar um sistema *job shop* de forma a produzir as quantidades demandadas, dentro de um horizonte de planejamento considerado de curto prazo (por exemplo: 24h), respeitando a relação de dependência entre *setups* e minimizando as perdas de tempos não produtivos.

A presente pesquisa demonstrou que é possível por meio do uso de técnicas matemáticas e abordagens da pesquisa operacional, traduzir problemas da realidade industrial de forma que as conclusões tiradas do modelo possam inferir decisões para o problema real.

Levando-se em consideração que o problema em questão é do tipo *NP-hard*, e por um bom tempo os algoritmos que poderiam achar soluções ótimas para estes problemas, não puderam existir, somente com os avanços em TI tornou-se possível os sistemas de *scheduling* baseado em computador (CONWAY; MAXWELL; MILLER, 2003; LEUNG, 2004; PINEDO, 2008). O algoritmo implementado em MATLAB, permitiu encontrar soluções ótimas ou realizáveis em um tempo de computacional baixo, o Quadro 20 apresenta os tempos médios computacionais para até 50 produtos (35 segundos).

Tendo como indicador a redução da perda dos tempos não produtivos, os resultados dos os experimentos realizados sugerem que o GILGOCW gerou soluções satisfatórias. No primeiro experimento os cortes não homogêneos indicam que houve o aproveitamento das linhas de forma inteligente, distribuindo o tempo de forma a produzir mais de um produto minimizando o tempo ocioso. No segundo experimento, a programação gerada otimizou a utilização das linhas por meio da não utilização da linha 6 e também minimizou a perda dos tempos não produtivos (*setup* e folga). No terceiro experimento, a programação gerada para o problema otimizou a utilização das linhas por meio da não utilização da linha 9 e também minimizou a perda dos tempos não produtivos (*setup* e folga), por meio da geração de cortes não homogêneos. As perdas de tempos não produtivos foram respectivamente 1.67%, 1.39% e 3.75%.

Vale ressaltar, que esta pesquisa se caracteriza como sendo uma pesquisa aplicada, com o desenvolvimento de um modelo. As considerações aqui feitas têm por finalidade relatar os fatos constatados nos experimentos realizados, visando contribuir com futuras pesquisas na área.

Assim, levando-se em consideração os resultados dos experimentos realizados pode-se responder à questão central desta pesquisa: que é possível o desenvolvimento de um modelo eficaz que permita sequenciar um sistema *job shop* de forma a produzir as quantidades

demandadas, dentro de um horizonte de planejamento considerado de curto prazo, respeitando a relação de dependência entre *setups* e minimizando as perdas de tempos não produtivos.

7.1 Recomendações para Estudos Futuros no Tema

Como recomendação para trabalhos futuros, duas sugestões podem ser colocadas. Em primeiro lugar, recomenda-se a pesquisa em ambientes onde a capacidade produtiva, por mais que tenha sido otimizada, é incapaz de atender a demanda. Nesse ambiente a equipe que calcula e acompanha a demanda dos produtos deverá sugerir algum fator de prioridade para o sequenciamento.

Em segundo lugar, sobre a questão do último produto feito no horizonte de tempo anterior, sugere-se que após a solução encontrada, considerando-se o *setup* oriundo da programação anterior igual a zero (o que indica quais produtos serão feitos em quais linhas), reordenar os produtos da linha de forma a achar o menor *setup* dependente e então subtrair-lo do tempo de *setup* da programação anterior.

Pode-se considerar também, com base no desenvolvimento do modelo proposto, em relação à identificação de sequenciamentos de menor *setup*, o aperfeiçoamento da rotina de ordenação de forma a obter-se soluções ótimas em um tempo computacional tão baixo quanto o gerado e com maior precisão na identificação da sequência com o menor *setup*, reduzindo ainda mais o número de iterações.

Para dar continuidade ao trabalho, pretende-se testar o modelo, em situações reais, com vista ao seu aprimoramento.

REFERÊNCIAS BIBLIOGRÁFICAS

ADAMS, J.; BALAS, E.; ZAWACK, D. The shifting bottleneck procedure for job shop scheduling. **Management Science**. v. 34, p. 391-401, 1988.

ALDAKHILALLAH, K.A.; RAMESH, R. Cyclic scheduling heuristics for a re-entrant job shop manufacturing environment. **International Journal of Production Research**.n.39, p. 2635–2657, 2001.

ALLAHVERDI, A.; GUPTA, J. N.D.; ALDOWAISAN, T.A review of scheduling research involving setup considerations. **Omega, Int. J. Mgmt Sci**. v. 27, p. 219-239,1999.

ALLAHVERDI, A.; NG, C.T.; CHENG, T.C.E.; KOVALYOV, M. Y.A survey of scheduling problems with setup times or costs. **European Journal of Operational Research**.n.187, p. 985–1032, 2008.

ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. **Pesquisa operacional**. Rio de Janeiro: Elsevier, 2007.

ARTIGUES, C.; BELMOKHTAR, S.; FEILLET, D.A new exact solution algorithm for the job shop problem with sequence dependent setup times. **In: Regin, J.C., Rueher, M. (Eds.), 1st International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems.** Lecture Note in Computer Science, v. 3011. Springer, p. 37–49, 2004.

ARTIGUES, C.; BUSCAYLET, F.A fast tabu search method for the job-shop problem with sequence-dependent setup times. **In: Proceedings of Metaheuristic International Conference**. Kyoto: Japan, August, p. 1–6, 2003.

ARTIGUES, C.; BUSCAYLET, F.; FEILLET, D. Lower and upper bounds for the job-shop scheduling problem with sequence dependent setup times. **In: Proceedings of the 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications.** New York: USA, July, p. 316–321, 2005.

ARTIGUES, C.; ROUBELLAT, O. An efficient algorithm for operation insertion in a multi-resource job-shop schedule with sequence dependent setup times. **Production Planning & Control.**n.13, v.2, p. 175–186, 2002.

AZIZOGLU, M.; WEBSTER, S. Scheduling job families about an unrestricted common due date on a single machine. **International Journal of Production Research**. n. 35, v.5, p.1321-1330, 1997.

BAKER, J. R.; McMAHON, G. B. Scheduling the general job-shop. **Management Science**. v. 31, p. 594-598, 1985.

BAKER, K. R. **Introduction to sequencing and scheduling**. New Jersey: Wiley, 1974. 305 p.

BAKER, K.R.; TRIETSCH, D. **Principles of sequencing and scheduling.** New Jersey: Wiley, 2009. 493 p.

BALAS, E., CERIA, S., CORNUÉJOLS, G., NATRAJ, N. Gomory cuts revisited. **Operations Research Letters**, v. 19, n.1, p. 1–9, 1996.

BALAS, E.; SIMONETTI, N.; VAZACOPOULOS, A. Job shop scheduling with setup times, deadlines and precedence constraints. **In: Proceedings of the 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications**. New York: USA, July. p. 520–532, 2005.

BALINSKI, M. L. Integer programming: methods, uses, computation. **Management Science**, v. 12, n. 3, p. 253-313, Nov. 1965.

BALLICU, M.; GIUA, A.; SEATZU, C. Job-shop scheduling models with set-up times. **Proceedings of the IEEE International Conference on Systems - Man and Cybernetics**. n. 5, p. 95–100, 2002.

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

BIEGEL, J. E.; WINK, L.J. Expert systems can do job shop scheduling: an exploration and a proposal. **Computers & Industrial Engineering** .n.1-4, v. 17, p. 347-352, 1989.

BROWN, A.; LOMNICKI, Z. A. Some applications of the branch of bound algorithm to the machine sequencing problem. **Operational Research Quartely**. v. 17, p. 173-186, 1966.

BRUCKER, P.; THIELE, O. A branch and bound algorithm for the general job shop with sequence dependent setup times. **OR Spektrum**. v. 18, 1996.

CANDIDO, M.A.B.; KHATOR, S.K.; BARCIA, R.M. A genetic algorithm based procedure for more realistic job shop scheduling problems. **International Journal of Production Research**, n.36, v.12, p. 3437–345, 1998.

CARLIER, J. The one-machine sequencing problem. **European Journal of Operation Research.**v. 11, p. 42-47, 1982.

CARLIER, J.; PINSON, E. An algorithm for solving the job shop problem. **Management Science**. v. 35, p. 164-176, 1989.

CHASE, R. B.; JACOBS, F. R.; AQUILANO, N. J. **Administração da produção para a vantagem competitiva.**Tradução R. Brian Taylor. 10. ed. Porto Alegre: Bookman, 2006. 724 p.

CHEUNG, W.; ZHOU, H. Using generic algorithms and heuristics for job shop scheduling with sequence-dependent setup times**. Annals of Operations Research**, n. 107, p. 65–81, 2002.

CHIZZOTTI, A. **Pesquisa em ciências humanas e sociais.** 7. ed. São Paulo: Cortez, 2005. 164 p.

CHOI, I.; CHOI, D. Alocal search algorithm for jobshop scheduling problems with alternative operations and sequence-dependent setups. **Computers & Industrial Engineering**. n.42, v.1, p. 43–58, 2002.

CHOI, I.C; KORKMAZ, O. Job shop scheduling with separable sequence-dependent setups.**Annals of Operations Research.** v. 70, p. 155-170, 1997.

CHUNG, D.; LEE, K.; SHIN, K.; PARK, J.A new approach to jobshop scheduling problems with due date constraints considering operation subcontracts. **Int. J. Production Economics.** v. 98, p. 238–250, 2005.

CLARKE, G.; WRIGHT, J.W. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. **Operations Research,** v. 11, p. 568-581, 1963.

CONWAY, R. W.; MAXWELL, W. L.; MILLER, L.W. **Theory of scheduling**. Mineola, N.Y: Dover Publications, 2003. 294 p.

COOPER, D. R.; SCHINDLER, P.S. **Métodos de pesquisa em administração**. Tradução Luciana de Oliveira da Rocha. 7. ed. Porto Alegre: Bookman, 2003. 640 p.

CORNUÉJOLS, G. Revival of the Gomory cuts in the 1990's. **Annals of Operations Research**, v. 149, n. 1, p.63–66, 2007.

CORRÊA, H. L; CORRÊA, C. A. **Administração de produção e operações**: manufatura e serviços: uma abordagem estratégica. 2. ed. São Paulo: Atlas, 2007. 690 p.

DAL GALLO, R. M. **Método heurístico eficiente para problemas de programação linear inteira com dimensão completa.** 2008**.** 55 f. Dissertação (Mestrado em Matemática Aplicada) – Curso de Pós-Graduação Matemática Aplicada,Universidade Estadual de Campinas**,** Campinas, 2008.

DANTZIG, G.B.; FULKERSON, D.R.; JOHNSON, S.M. Solution of a large scale travelling salesman problem. **Journal of the Operation Research Society of America**, v. 2,p. 393-410, 1954.

DAUZÈRE-PÉRÈS, Stéphane; LASSERRE, Jean-Bernard. On the importance of sequencing decisions in production planning and scheduling. **International Transactions in Operational Research**. p. 779–793, Sep. 2002.

DEFERSHA, F.M.; CHEN, M.A parallel genetic algorithm for a flexible job-shop Delivery Points.**Int J Adv Manuf Technol,**v. 49, p. 263-279, 2010.

ESPINOZA, D. G. Computing with multi-row Gomory cuts. **Operations Research Letters**. v. 38, n.1, p. 115–120, 2010.

FERNANDES, F. C. F.; GODINHO FILHO, M. **Planejamento e controle da produção:** dos fundamentos ao essencial. São Paulo: Atlas, 2010. 275 p.

FLICK, U. **Uma introdução à pesquisa qualitativa**. Tradução Sandra Netz. 2. ed. Porto Alegre: Bookman, 2004. 312 p.

FLYNN, B.B. **Group technology vs. process layout**: a comparison using computerized job shop simulation. Ph.D. Thesis, Indiana University, 1984.

FOCACCI, F.; LABORIE, P.; NUIJTEN, W. Solving scheduling problems with setup times and alternative resources. **In: Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling**. Breckenbridge, Colorado: USA. p. 92–101, 2000.

FONSECA, S. L. A. **Um Estudo Computacional de Cortes Derivados do Corte Chvátal-Gomory para Problemas de Programação Inteira.** 2007**.** 91 f. Dissertação (Mestrado em Engenharia Elétrica) – Curso de Pós-Graduação Engenharia Elétrica,Universidade Estadual de Campinas**,** Campinas, 2007.

GAITHER, N.; FRAZIER, G.**Administração da produção e operações**. Tradução José Carlos Barbosa dos Santos. 8. ed. São Paulo: Pioneira Thomson Learning, 2005. 598 p.

GANTT, H.L. **Organizing for Work.** New York: Harcourt, Brace, and Howe, 1919.

GANTT, H.L. Work, Wages, and Profits. **Engineering Magazine Co**. 2rd.New York, 1916.

GILMORE, P.C.; GOMORY, R.E.A linear programming approach to the cutting-stock problem. **Operations Research,** v. 9, n. 6, p. 849–859, 1961.

GILMORE, P.C.; GOMORY, R.E. A linear programming approach to the cutting-stock problem: Part II. **Operations Research,** v. 11, p. 863–888, 1963.

GLOVER, F.; SHERALI, H. D. Chvatal-Gomory-tiers cuts for general integer programs.**Discrete Optimization.**v.2, p. 51–69, 2005.

GOMORY, R.E. Outline of an algorithm for integer solutions to linear programs. **Bulletin of the American Mathematical Society.** n. 64, p. 275–278, 1958.

GRAHAM, R.L.; LAWLER, E.L.; LENSTRA, J.K.; KAN, A.H.G.R. Optimization and approximation in deterministic sequencing and scheduling: a survey, **Annals of Discrete Mathematics**. v.5, p. 287–326, 1979.

GUPTA, S.K. n jobs and m machines job-shop problems with sequence dependent setup times. **International Journal of Production Research**. n.20, v. 5, p.643-656, 1982.

HAESER, G.; GOMES–RUGGIERO, M. Aspectos teóricos de simulated annealing e um algoritmo duas fases em otimização global. **Tend. Mat. Apl. Comput**. n. 3, v.9, p. 395-404, 2008.

HAYES, R.; PISANO, G.; UPTON, D.; WHEELWRIGHT, S. **Produção, estratégia e tecnologia**: em busca da vantagem competitiva. Tradução Marcelo Klippel. Porto Alegre: Bookman, 2008. 384 p.

HEMMONDHAROP, P. **A comparative study of just-in-time (JIT) and theory of constraints (TOC) systems with varying constraint locations and operacional characteristics.**2001. 135 f. Tese (Doutorado em Administração) – Curso de Pós Graduação em Administração da Produção, Universidade do Missouri, Missouri, 2001.

HERRMANN, J. W. (Ed.). **Handbook of production scheduling**. New York: Springer,2006. 318 p.

HERSHAUER, J.C. **An empirically-derived simulation to explore sequencing decisions**. Ph.D. Thesis, Indiana University, 1970.

HERTZ, A.; WIDMER, M. An improved tabu search approach for solving the job shop scheduling problem with tooling constraints. **Discrete Applied Mathematics**. n. 65, p. 319–345, 1996.

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**.Tradução Ariovaldo Griesi. Porto Alegre: AMGH, 2010. 852 p.

JACKSON, R. J. An extension of Johnson´s results on job lot scheduling. **Naval Research Logistics Quarterly**, 3, 201-203, 1956.

JACKSON, R. J. Scheduling a production line to minimize maximum tardiness. **Research Report 43, Management Science Research Project**. University of California, Los Angeles, 1955.

JOHNSON, E. L. IFORS' Operational Research Hall of Fame: Ralph E. Gomory. **International Transactions in Operational Research**.n.12, p. 539-543, 2005.

JOHNSON, L. A.; MONTGOMERY, D. C. **Operations research in production planning, scheduling and inventory control.**New York: John Wiley, 1974.

JOHNSON, S. M. Optimal two and three-stage production schedules with setup times included. **Naval Research Logistics Quarterly,** 1, 61-67, 1954.

JONSSON, P; MATTSSON, S. The implications of fit between planning environments and manufacturing planning and control methods**. International Journal of Operations & Production Management**. v. 23, n.8, p. 872-900, 2003.

JÜNGER, M.; LIEBLING, T.M.; NADDEF, D.; NEMHAUSER, G.L.; PULLEYBLANK, W.R.; REINELT, G.; RINALDI, G.; WOLSEY, L. A. (Ed.). **50 years of integer programming 1958-2008**: from the early years to the state-of-the-art. York : Springer, 2010. 804 p.

KIM, S.C.; BOBROWSKI, P.M. Impact of sequence-dependent setup times on job shop scheduling performance. **International Journal of Production Research**. n. 32, v.7, p. 1503-1520, 1994.

KIM, S.C.; BOBROWSKI, P.M. Scheduling jobs with uncertain setup times and sequence dependency.**Omega**.n.25, v.4, p. 437–447, 1997.

KÖPPE, M.; LOUVEAUX, Q.; WEISMANTEL, R.; WOLSEY, L. A. Extended formulations for Gomory corner polyhedral. **Discrete Optimization**. v. 1, p.141–165, 2004.

KRAJEWSKI, L.; RITZMAN, L.; MALHOTRA, M. **Administração de produção e operações**. Tradução Miriam Santos Ribeiro de Oliveira.8. ed.São Paulo: Pearson Prentice Hall, 2009. 615 p.

LACHTERMACHER, G. **Pesquisa operacional na tomada de decisões**. 4. ed.São Paulo: Pearson Prentice Hall, 2009. 223 p.

LETCHFORD, A. N; LODI, A. Strengthening Chavátal-Gomory cuts and Gomory fractional cuts. **Operational Research Letters**. v. 30, p.74−82, 2002.

LEUNG, J. Y-T. **Handbook of scheduling**: algorithms, Models, and performance analysis. Boca Raton: Chapman & Hall/CRC, 2004. 1224 p.

LOMNICKI, Z. A. A branch and bound algorithm for the exact solution of the three-machine scheduling problem. **Operational Research Quartely**. v. 16, p. 89-100, 1965.

LOW, C.; HSU, C. M.; HUANG, K.I. Benefits of lot splitting in job-shop scheduling. **International Journal of Advanced Manufacturing Technology**. n. 24, p. 773–780, 2004.

LOW, C.Y. Job shop scheduling heuristics for sequence-dependent setups. **Computers & Industrial Engineering**.n.29, p. 279-283, 1995.

LUH, P.B.; GOU, L.; ZHANG, Y.; NAGAHORA, T.; TSUJI, M.; YONEDA, K.; HASEGAWA, T.; KYOYA, Y.; KANO, T. Job shop scheduling with group-dependent setups, finite buffers and long time horizon. **Annals of Operations Research: Mathematics of Industrial Systems**. n.76, p. 233–259, 1998.

MACCARTHY, B. L.; LIU, Jiyin. Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. **International Journal of Production Research**, v. 31, n.1, p. 59-79, Jan. 1993.

MARKOWITZ, H.M.; MANNE, A. S.On the solution of discrete programming problem. **Econometrica**, v. 25, p. 84-110, 1957.

MARQUES, F. P.; ARENALES, M. N. O problema da mochila compartimentada e aplicações. **Pesquisa Operacional**.v. 22, n.3, p. 285–304, 2002.

MARTINS, P. G.; LAUGENI, F. P. **Administração da produção.** São Paulo: Saraiva, 2005. 562p.

MASON, S.J.; FOWLER, J.W.; CARLYLE, W.M. A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. **Journal of Scheduling.**n.5, v.3, 247–262, 2002.

MASON, S.J.; FOWLER, J.W.; CARLYLE, W.M.; MONTGOMERY, D.C. Heuristics for minimizing total weighted tardiness in complex job shops. **International Journal of Production Research**. n.43, p. 1943–1963, 2005.

MORABITO, R. Modelos de otimização para o problema de corte nas indústrias de papel e papelão e de móveis. **Gestão & Produção.**v. 1, n.1, p. 59–76, 1994.

MOREIRA, D. A. **Administração da Produção e Operações**. São Paulo: Pioneira Thomson Learning, 2004. 619 p.

MOREIRA, D. A. **Pesquisa operacional:** curso introdutório. 2. ed. São Paulo: Cengage Learning, 2010.

NADERI, B.; GHOMI, S.M.T. F.; AMINNAYERI, M. A high performing metaheuristic for job shop scheduling with sequence-dependent setup times. **Applied Soft Computing**. v. 10, p. 703–710, 2010.

NAGANO, M. S.; BRANCO, F. J. C.; MOCCELLIN, J. V. Soluções de alto desempenho para a programação da produção flow shop. **GEPROS – Gestão da produção, operações e sistemas.**n. 2, p. 11-23, Abr-Jun/2009.

NEMHAUSER, G. L.; WOLSEY, L. **Integer and combinatorial optimization**. New York : Wiley, 1999. 763 p.

O'GRADY, P.J.; HARRISON, C. Search based job scheduling and sequencing with setup times. **Omega Int J Manage Sci**. n. 16, p. 547-552, 1998.

OVACIK, I.M.; UZSOY, R. Exploiting shop floor status information to schedule complex job shops.**Journal of Manufacturing Systems.**n.13, p.73-84, 1994.

PACHECO, R. F.; SANTORO, M. C. Proposta de classificação hierarquizada dos modelos de solução para o problema de job shop scheduling. **Gestão e Produção**. v. 6, n.1, p. 1–15, 1999.

PALMER, R. D. **Maintenance Planning and Scheduling Handbook.**2. ed. New York : McGraw-Hill, 2006. 820 p.

PATTERSON, M.C. Analysis of setup time at constraint resources. **Int J Prod Res**.v.31, p.845-849, 1993.

PILEGGI, G. C. F.; MORABITO, R.; ARENALES, M. N. Heurísticas para os problemas de geração e seqüenciamento de padrões de corte bidimensionais. **Pesquisa Operacional.** v. 27, n.3, p. 549–568, 2007.

PINEDO, M. L. **Scheduling: theory, algorithms, and systems**. 3rd. New York: Springer, 2008. 671 p.

POLDI, K. C.; ARENALES, M. N. Heurísticas para o problema de corte de estoque unidimensional inteiro. **Pesquisa Operacional.**v. 26, n.3, p. 473–492, 2006.

POLDI, K. C.; ARENALES, M. N. O problema de corte de estoque unidimensional multiperíodo. **Pesquisa Operacional.**v. 30, n.1, p. 153–174, 2010.

RAGSDALE, C. T. **Modelagem e análise de decisão.** Tradução Luciana Penteado Miquelino. São Paulo: Cengage Learning, 2009. 590 p.

RANGEL, S.; FIGUEIREDO, A. G. O problema de corte de estoque em indústrias de móveis de pequeno e médio portes. **Pesquisa Operacional**. v. 28, n.3, p. 451–472, 2008.

REIS, J. **Uma introdução ao scheduling**. Instituto Superior de Ciências do Trabalho e da Empresa – ISCTE, 1996.

RUSSELL, R. S.; TAYLOR III, B. W. **Operations management**: quality and competitiveness in a global environment. 5rd. New Jersey: Willey, 2006. 808 p.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. **Administração da produção**.Tradução Henrique Luiz Corrêa.3. ed. São Paulo: Atlas, 2009. 728 p.

SMITH, W. E. Various optimizers for single stage production. **Naval Research Logistics Quarterly**, 3, 59-66, 1956.

SOBRAL, F.; PECI, A. **Administração**: teoria e prática no contexto brasileiro. São Paulo: Pearson Prentice Hall, 2008. 398 p.

SOTOSKOV, Y.N; TAUTENHAHN, T.; WERNER, F. On the application of insertion techniques for job shop problems with batch setup times. **RAIRO Oper Res**, 1998.

SUN, J.U.; YEE, S.R.; HWANG, H. Job shop scheduling with sequence dependent setup times to minimize makespan. **International Journal of Industrial Engineering - Theory Applications and Practice**.n.10, v.4, p. 455–461, 2003.

SUN, X.; NOBLE, J.S. An approach to job shop scheduling with sequence-dependent setups. **Journal of Manufacturing Systems**.n.18, p. 416–430, 1999.

TAHAR, D.N.; YALAOUI, F.; AMODEO, L.; CHU, C. An ant colony system minimizing total tardiness for hybrid job shop scheduling problem with sequence dependent setup times and release dates. **In: Proceedings of the International Conference on Industrial Engineering and Systems Management.** Marrakech: Morocco. May, p. 469–478, 2005.

TARANTILIS, C.D.; KIRANOUDIS,C.T. A list-based threshold accepting methodfor job shop scheduling problems. **Int. J. Production Economics.** v. 77, p. 159–171, 2002.

TUBINO, D. F. **Planejamento e controle da produção:** teoria e prática. São Paulo: Atlas, 2008. 190 p.

WILBRECHT, J.K.; PRESCOTT, W.B. The influence of setup time on job shop performance. **Manage Sci**. v.16, p. 274-280, 1969.

WOLSEY, L. A. **Integer programming**. New York : Wiley, 1998. 264 p.

ZAMBELLI, G. On degenerate multi-row Gomory cuts. **Operations Research Letters**. v. 37, p. 21–22, 2009.

ZHOU, C.; EGBELU, P.J. Scheduling in a manufacturing shop with sequence-dependent setups. **Robotics Comput-Integr Manufact** .v.5, p.73-81, 1989.

ZHU, X.; WILHELM, W. Scheduling and lot sizing with sequence-dependent setup: Ascheduling problem with sequence dependent setups. **Int J Adv Manuf Technol**. v.49, p. 263–279, 2006.

ZOGHBY, J.; BARNES, J.W.; HASENBEIN, J.J. Modeling the reentrant job shop scheduling problem with setups for metaheuristic searches. **European Journal of Operational Research**.n.167, p. 336–348, 2005.

APÊNDICE 01 – ROTINAS EM MATLAB

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% METODO DE GERAÇÃO DE COLUNAS UTILIZANDO GILMORE E GOMORY
% CLARKE & WRIGHT - PARA ORDENAÇÃO DA MOCHILA
% OBJETIVO: MINIMIZAR A PERDA ATENDENDO A DEMANDA CONSIDERANDO SETUP
% (DEPENDENTE)
% VARIÁVEIS
% L = Tamanho máximo da barra (horizonte de planejamento)
% l = Tamanho dos cortes (lotes)
% d = Demanda
% A = Matriz de Construção
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

formatshortg;
% Entradas
% Inserir o valor de L
% Inserir os valores li sendo i o indexador variando de 1 m(colunas)
% Inserir os valores de di sendo i o indexador variando de 1 a n(linhas)

%% Passo 1 - Entrada de dados
L=24;
d=[26 8 12 4 2];
l=[2 3 4 6 9];
set_max=0.4;

%% Setup Randômico - gera os valores da matriz dos setups dependentes
rnd_a=(rand(length(l)));
 rnd_a=rnd_a*(48/(1/L));
 rnd_b=max(rnd_a);
 rnd_c=max(rnd_b)
if (max(rnd_c)>set_max)
     cut_set=(max(rnd_c)/set_max)
end
 rnd_a=rnd_a/cut_set;
 m_ones=ones(length(l));
 m_eye=eye(length(l));
num=(m_ones - m_eye).*rnd_a;
disp('Matriz Setup')
disp(num)

%% Atribuições
ST=num;
type=1; % 1 equivale a <= e -1 equivale a >=
iteracao=1;% inicializa a quantidade de iterações
limite = 3000;% limite de iterações para solução do problema
rel=0;
xb_aux=[];

%% Passo 2 - Montagem da Matriz A
[B, N, A, b]= Montamatriz(L, l, d, type);% monta a matriz A
invB=inv(B);% gera a inversa de B
[n,m]= size(B);% n linhas x m colunas
vk=zeros(1,m);% vetor da mochila

%% Passo 3 - Identificar vetores de índice e coef de f(x)
cn=m+1; %coef nãobásico
cb=(1:m); %coef básico
cf=ones(1,m);%coef f(x)
```

```
%% Clark & Wright
monta_par=0;
v_work=[];%matriz de trabalho para montagem das sequências
final=[];%matriz final com as sequência
pula=0;%flag para apagar repetido espelhado
matrix_tempos =num;
[m_mt,n_mt] = size(matrix_tempos);% carregada m da matriz tempo e n da
matriz tempo
matrix_decide=zeros(m_mt,3);%monta entrada e saída
temp_sort=zeros(m_mt,3);%transforma matriz em vetor
for i=1:m_mt
for j=1:n_mt
if(i~=j)
monta_par=monta_par+1;
temp_sort(monta_par,1)=matrix_tempos(i,j);
temp_sort(monta_par,2)=i;
temp_sort(monta_par,3)=j;
end
end
matrix_sort=sortrows(temp_sort,1);%ordena do menor para o maior
end
confere=matrix_sort;
i = 1;
duo=zeros((m_mt-1),3);
tic
for i=1:size(matrix_sort,1);
arranjo = matrix_sort(i,:);
v_work=arranjo;
entrada=(matrix_sort(i,2));
saida=(matrix_sort(i,3));
for t=1:(m_mt-2)
busca=find(matrix_sort(:,2)==saida);
for j=1:(m_mt-1)
duo(j,:)= matrix_sort(busca(j),:);
end
for j=1:(length(entrada))
kill=find(duo(:,3)==entrada(j));
duo(kill,:)=[];
end
saida=duo(1,3);
entrada=[entrada duo(1,2)];
v_work=[v_work; (duo(1,:))];
duo=zeros((m_mt-1),3);
busca=[];
end
soma= sum(v_work(:,1));
    semi_final=[v_work(:,2); v_work(size(v_work,1),3)].';
semi_final=[semi_final soma];
final=[final; semi_final];
end
final_sort=sortrows(final,size(final,2));
final_sort(:,size(final,2))=[];

%% Reordena
disp('Início')
disp(datestr(now))
v=linspace(1,m,m);% vetor inicial de 1 até m
vp=final_sort;
[nvp,mvp]=size(vp);
vpa=zeros(1,mvp-1);% base para somatório
vpx=zeros(nvp,1);% armazenamento do resultado
```

```
for i= 1:nvp
for j= 1:mvp-1
vpa(j,i)=(ST(vp(i,j),vp(i,(j+1))));%arranjo para soma
end
vpx(i)=(sum(vpa(:,i)));% matriz com os somatórios
end
vpidx=linspace(1,(length(vp(:,1))),(length(vp(:,1))));
%% Looping
for iteracao=1 : limite % loop para iteração
if rel == 0
disp('Iteracao');
disp(iteracao);

%% Passo 4 - Calcular a SoluçãoBásica
xb=invB*b; %xb recebe a soluçãobásica
disp('xb')
disp(xb)
if (min(xb)<0)% testa a soluçãobásica (se<0 FIM com nova entrada)
disp('xb não respeitou a condição de não negatividade!')
xb=xb_aux;
iteracao=iteracao-1;
    [pcorte, lcorte, mcorte, ncorte]=relatorio(xb, invB, l, L, b, d,
iteracao);%m, cf,
rel=1;
disp(datestr(now));
disp('Tempo gasto no cálculo');
toc
break
end

%% Passo 5 - Vetor Multiplicador Simplex
vm=cf*invB;% coef f(x)*inversa de B

%% Passo 6 - Calculo de vk (vetor da mochila "knapsack")
[dm, dmo, nk, mk, si, sit, Nm, ST, vp, vpok]= Mochila(L, N, l, cn, cb, vm,
ST, m, vpidx,vp, iteracao);
%Retornou a solução Inicial
fim=0;
fmax=0;

while fim == 0
[fmax, si]= Mochila_testa(dmo, mk, sit); %Calcula o retorno si,
[icmz]= Mochila_cmz(sit, mk); %Encontra primeiro coef > 0
fmm=0;
[fmm, sit]= Mochila_fmm(sit, L, dmo, fmax, icmz);% Decide se há melhor
retorno
[icmz]= Mochila_cmz(sit, mk); %Encontra o próximo coef > 0
if fmm == 0
[fim]= Mochila_final(icmz, sit);%Verifica se há outras posições no vetor
else
aux=0;
for i=1:icmz
aux=aux+(sit(i)*dmo(2,i));
end
sit(icmz+1)=floor(L-aux)/dmo(2,icmz+1);
aux=0;
for i=1:mk-1%trocou o dmo para teste
aux=aux+(sit(i)*dmo(2,i));
end
sit(mk-1)=floor(L-aux)/dmo(2,mk-1);
fim=1;
```

```
end
end% do while

for i=1:mk % Vetor da Mochila
kk(1,dmo(3,i))=si(1,i);

end
for i=1:mk-1
vk(1,i)=kk(1,i);
end
vk=vk.';
crel=(dmo(2,mk))- vm * vk;
if (crel < 0 & (abs(crel)>eps))
    [ind, DS, AP]=D_Simplex(invB, vk, m, xb);
fprintf('A coluna que sai é X %.0d    \n\n', ind);
disp('Nova invB')
    [invB]=Atualiza(DS, invB, cb, m, ind);
disp(invB);
else
%imprime
    [pcorte, lcorte, mcorte, ncorte]=relatorio( xb, invB, l, L, b, d,
iteracao);%m, cf,
beep
rel=1;
toc
end
end% relatório
vk=vk.';
vk=zeros(1,m);
xb_aux=xb;
end

%% monta matriz identidade (cortes homogêneos)
function [B, N, A, b]=Montamatriz(L, l, d, type)
mid=eye(length(d));
N=mid(:,2);
vL = floor(L*(l.^-1));% L/l
B = diag(vL);% monta matriz identidade (cortes homogêneos)
A=[B N];% concatenação de B e N em A
b = d(:);% monta vetor b (quantidades)

%% Função mochila – retorna a solução inicial com o somatório do setup
function [dm, dmo,nk, mk, si, sit, Nm, ST, vp, vpok]= Mochila(L, N, l, cn,
cb, vm, ST, m, vpidx,vp, iteracao)
%% montagem de dm
Nm=[N(1);N(2);N(3)];
dm=[vm; l];%dados da mochila dm (1a linha vm e 2a linha cortes)
dm=[dm; cb];%resultado da concat anterior com cb na 3a linha
dm=[dm,Nm];%concat dm com enésima coluna: folga (N)
[nk,mk]=size(dm);
dm(nk,mk)=cn; % Dados da mochila inserido o coef nãobásica
disp('Dados da Mochila');
disp(dm);

%% calculo da mochila ordenada
div=zeros(1,mk-1);%vetor com zeros
for i= 1:mk-1
    div(i)= dm(1,i)/dm(2,i);% divisão dos elementos da 1a com a 2a linhas
end
[res,idx]=sort(div);% res=ordenação da divisão , idx=ordenação do índice do
</>
```

```
xres=res(end:-1:1);% inversão da ordenação do res
xidx=idx(end:-1:1);% inversão da ordenação dos índices
for i=1:mk-1
dmo(:,i)=[dm(:,xidx(i))];%montagem de dmo
end
dmo(:,mk)=dm(:,mk);%Mochila Ordenada
disp('Mochila Ordenada');
disp(dmo);

%% Reordena
vpok=(vp(vpidx(iteracao),:));% sequência escolhida
%disp(iteracao);
fprintf('Iteração Número: %d \n',(iteracao));
fprintf('Índice do arranjo na matriz não ordenada (vp): %d
\n',(vpidx(iteracao)));
disp('Arranjo escolhido:')
disp(vpok);
dmro=dmo;

for i=1: length(vpok)
    [zn, zi]=find(dmo(3,:)==vpok(i));
dmro(:,i)=(dmo(:,zi));
end

disp('dmo reordenado')
disp(dmro);

%% Setup
for i= 1:mk-2
S1=(dmro(3,i));
    S2=(dmro(3,(i+1)));
dmro(2,(i+1))= dmro(2,(i+1))+(ST(S1,S2));
end
disp('Mochila com SETUP');
disp(dmro);

%% si = Solução Inicial
dmo=dmro;
aux=L;
si=zeros(1,mk);
for i=1:mk% montagem do vetor
si(i)=floor(aux/dmo(2,i));
aux = aux-(si(i)*dmo(2,i));
end
fmax=0;%Retorno Maximo da mochila
sit=si;%SoluçãoInicial (vetor de trabalho)

%% Encontra o primeiro coef maior que 0 da DIR para ESQ da solução inicial
function[icmz]= Mochila_cmz(sit, mk)
icmz=0;
i=mk-1;
while icmz == 0
if sit(i)> 0
icmz=i;
end
if i > 1
    i=i-1;
end
end
```

```
%% Calcula uma segunda opção de mochila
function [fmm, sit]= Mochila_fmm(sit, L, dmo, fmax, icmz)
sit(icmz)=sit(icmz)-1;
tm=0;
for i=1:icmz
tm=tm+(sit(i)*dmo(2,i));
end
divide=(L-tm)/(dmo(2,icmz+1));
sit(icmz+1)=floor(divide);
aux=0;
for i=1:icmz
aux=aux+(dmo(1,i)*sit(i));
end

%% compara o resultado com fmax: se > fmm=1 se < fmm=0
if (dmo(1,icmz+1)*sit(icmz+1)+aux)>fmax
fmm=1;
else
fmm=0;
end

%% Calcula o retorno máximo
function [fmax, si]= Mochila_testa(dmo, mk, sit)
tm=0;%Somatório do retorno
for i=1:mk-1
tm= tm +(dmo(1,i)*sit(i));% ∑ da multiplicação da 1a linha da dmo pela
solucao inicial da mochila
end
si=sit;
fmax=tm;% retorno Maximo

%%Verifica se todo vetor da mochila foi lido
function [fim]= Mochila_final(icmz, sit)
temp=icmz;
icmz=0;
while icmz == 0
if sit(temp) > 0
icmz=temp;
end
temp=temp-1;
if temp == 0
icmz=9999999999;
end
end
if icmz == 9999999999
fim=1;
else
fim=0;
end

%% Calcula direção simplex
function [ind, DS, AP]=D_Simplex(invB, vk, m, xb)
DS=(-1*invB)*vk;
if (max(DS) == 0 & min(DS) == 0)% exclui possibilidades para todos = 0;
disp('Não tem solução ÓTIMA')
rel=1; %sai do programa
else
for i=1:m
if xb(i) == 0  & DS(i) == 0
AP(i)= inf;
else
```

```
AP(i)=-xb(i)/DS(i);

if (AP(i)) == -inf
AP(i) = inf;
end
if (AP(i)) <= 0
AP(i) = inf;
end
end
end

[minimo,ind]=min(AP);
End

%% Atualiza a matriz A para nova iteração
function [invB]=Atualiza(DS, invB, cb, m, ind)

for j=1:m
       invB(ind,j)=(-1/(DS(cb(ind)))*invB(ind,j));
for i=1:m
if i ~= ind
       invB(i,j)=invB(i,j)+(DS(cb(i))*invB(ind,j));
end
end
end

%% relatório final
function [pcorte, lcorte, mcorte, ncorte]= relatorio(xb, invB, l, L, b, d,
iteracao)
r_final=[];
soma_linha=[];
soma_coluna=[];
pcorte=inv(invB);
[mcorte,ncorte]=size(pcorte);
lcorte=l.';
fprintf('O horizonte de planejamento é: %.0f horas.\n', L);
disp('Solução Gerada')
disp('=========================================================================
')
solucao=xb.';

for f=1: length(solucao)
r_final=[r_final pcorte(:,f).*solucao(1,f)];
end
for f=1: length(r_final)
soma_linha=[soma_linha;(sum(r_final(f,:)))];
end
for f=1: length(r_final)
soma_coluna=[soma_coluna;(L - (sum(l.'.*pcorte(:,f))))];
end

sobra=soma_coluna.';


fprintf('Número de Iterações: %.2f \n\n', iteracao);

disp('Relatório Final')
disp('=========================================================================
')
for i= 1: length(l)
```

```
    time_find=find(pcorte(:,i));
fprintf('A linha %.0f', i);
multi=0;
for j=1 :length(time_find)
if(solucao(:,i))>0
multi= multi + ((pcorte(time_find(j,1),i))*l(1,time_find(j,1)));
            fprintf(' produziu %.0f lote(s) do produto %.0f,',
pcorte(time_find(j,1),i),(time_find(j,1)));
else
fprintf(' NÃO FOI UTILIZADA')
sobra(1,i)=0;
end
end
if(solucao(:,i))>0
fprintf(' em %.0f hora(s) ', multi);
fprintf('com uma perda de %.0f hora(s).\n',L-multi);
else
fprintf('\n')
end
end
disp('=====================================================================
')
perda=((sum(sobra)/(L*length(l)))*100);
fprintf('O percentual de perda será de: %.2f \n', perda);

disp('=====================================================================
')
fprintf('Para atender a demanda: \n');
for i=1:length(l)
fprintf('A linha %.0f --> Repetir a programação %.0f vezes. \n', i
,solucao(1,i));
end

disp('=====================================================================
')
soma_linha=[0; soma_linha; 0];
elinho=[0; l.'; 0];
maximus=[solucao; pcorte; sobra];
maximus=[elinho maximus soma_linha];
disp(maximus)

disp('=====================================================================
')
```