

UNIVERSIDADE DO VALE DO RIO DOS SINOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
APLICADA

Gilberto Irajá Müller

Utilização da Busca Tabu para a
Geração de um Modelo Aplicado ao
Job-shop Scheduling Problem Considerando
um Sistema de Manufatura Flexível

São Leopoldo

2006

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Gilberto Irajá Müller

Utilização da Busca Tabu para a Geração de um Modelo Aplicado ao
Job-shop Scheduling Problem Considerando um Sistema de
Manufatura Flexível

Dissertação apresentada à Universidade do Vale do Rio dos Sinos (UNISINOS) como requisito parcial para a obtenção do título de Mestre em Computação Aplicada.

Orientador: Prof. Dr. Arthur Tórgo Gómez

São Leopoldo

2006

Ficha catalográfica elaborada pela Biblioteca da
Universidade do Vale do Rio dos Sinos

M958u Müller, Irajá Gilberto
Utilização da busca tabu para a geração de um modelo aplicado
ao Job-shop scheduling problem considerando um sistema de manufatura
flexível / por Gilberto Irajá Müller. – 2006.
128 f. : il. ; 29cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos,
Programa de Pós-Graduação em Computação Aplicada, 2006.

“Orientação: Prof. Dr. Arthur Tórgo Gómez, Ciências Exatas e
Tecnológicas”.

1. Engenharia de software. 2. Sistema de Manufatura flexível. 3. Job
Shop. 4. Algoritmo busca tabu. I. Título.

CDU 004.41

Catálogo na Publicação:
Bibliotecária Eliete Mari Doncato Brasil - CRB 10/1184

Gilberto Irajá Müller

“Utilização da Busca Tabu para a Geração de um Modelo Aplicado ao
Job-shop Scheduling Problem Considerando um Sistema de
Manufatura Flexível”

Dissertação apresentada à Universidade do Vale do Rio dos Sinos (UNISINOS) como
requisito parcial para obtenção do título de mestre em Computação Aplicada.

Aprovada em fevereiro de 2006

BANCA EXAMINADORA

Prof. Dr. Arthur Tórgo Gómez

Prof. Dr. José Vicente Canto dos Santos

Prof. Dr. Luiz Antonio Nogueira Lorena

“Em momentos de crise, só a imaginação é mais importante que o conhecimento.”
(Albert Einstein)

Dedicatória

Dedico este trabalho para as pessoas que fazem a minha vida ter sentido:
Belmonte Müller
Elcira Corrêa de Moraes
Cátia Simone Robinson Müller

Agradecimentos

Quero agradecer as seguintes pessoas e instituições que de alguma maneira me ajudaram ao longo destes dois anos:

- Aos meus pais pelo apoio nutricional;
- Aos amigos que fiz nestes dois anos: Glauco, Leonardo, Scopel, Sérgio Crespo e Luna Lunar. Não esquecerei da churrasqueira;
- Ao tchê Arthur Tórigo Gómez, pela orientação e acima de tudo, pelo conhecimento adquirido;
- A CAPES e o MEC pelo apoio financeiro;
- A minha esposa Cátia Simone Robinson Müller por ter ajudado nos momentos de fraqueza, sono e irritação;
- A todos que, de alguma maneira, colaboraram para a realização deste trabalho.

Resumo

Este trabalho tem como objetivo a geração de um modelo de escalonamento aplicado ao *Job-shop Scheduling Problem* num Sistema de Manufatura Flexível que considera o tempo total de produção (*makespan*), o tempo total de atraso, o tempo total parado e o tempo total ocioso. O modelo proposto é composto por: (a) uma função objetivo que reflete, através de suas variáveis de decisão e seus pesos respectivos, as estratégias de otimização, e de (b) uma arquitetura que está dividida em cinco fases. O modelo utilizou o algoritmo Busca Tabu que, através de duas estratégias de geração de vizinhanças, busca a otimização da função objetivo. A arquitetura do modelo baseia-se na extração da demanda de produção, na Tecnologia de Grupo, nas Regras de Despacho, no Algoritmo Busca Tabu e na gravação do plano de produção, para tratar os Problemas de Seleção de Partes (Famílias de Partes) e do Escalonamento.

Foram realizados, através de um estudo de caso, diversos experimentos que possibilitaram a comparação de estratégias de otimização em relação ao escalonamento real utilizado, e a constatação de conflitos em relação às variáveis de decisão. Para a validação do modelo foram utilizados trabalhos clássicos que propõem a solução do *Job-shop Scheduling Problem*.

Palavras-chave: Tecnologia de Grupo, Famílias de Partes, Busca Tabu, *Job-shop*, Sistema de Manufatura Flexível.

“Using Tabu Search for the Generation of Model Applied *Job-shop Scheduling Problem* Considering a Flexible Manufacturing System”

Abstract

This paper has the aim of generating a scheduling model applied to Job-shop Scheduling Problem in Flexible Manufacturing System, which considers the makespan, total tardiness time, total stop time, total idle time. The model proposed is composed for: (a) an objective function that reflects, through its variables of decision and its weights, the optimization strategies, and (b) architecture that is divided in five phases. The model used the Tabu Search algorithm which, through two strategies neighborhoods generation, searching the objective function optimization.

The model architecture is based on extraction of production demand, in the Group Technology, in the Dispatching Rules, in the Tabu Search algorithm and save production plan, to deal the Part Selections (Part Families) and Scheduling Problems.

Through a study of case, it has been realized several experiments which makes it possible the comparison of optimization strategies and real scheduling, and which proves conflicts in decision variables. For model validation it has been used classic works which propose the solution of Job-Shop Scheduling Problem.

Keywords: Group Technology, Part Families, Tabu Search, Job-shop, Flexible Manufacturing System.

Lista de Figuras

2.1 - Posicionamento dos SM em um Sistema de Produção	18
2.2 - (a) Roteamento variado e (b) Roteamento fixo.....	20
2.3 - Classificação dos SM quanto ao Nível de Automação	22
2.4 - (a) Flexibilidade e Produtividade e (b) Complexidade nos SM.....	23
3.1 - Partes com durezas diferentes	27
3.2 - Partes com processos similares e atributos diferentes.....	27
3.3 - Esquema de codificação do modelo Opitz	33
3.4 - Código para cada dígito do modelo Opitz.....	34
3.5 - Projeto da parte para o modelo Opitz.....	34
3.6 - Projeto da parte para o modelo MULTICLASS.....	35
3.7 - Esquema de codificação MULTICLASS.....	35
3.8 - Esquema de codificação do modelo KK-3.....	36
3.9 - (a) Matriz inicial e (b) Matriz após agrupamento	37
4.1 - Relação das categorias com a Flexibilidade.....	45
4.2 - (a) CNC horizontal e (b) CNC sem a proteção (4 eixos)	46
4.3 - Computador numa célula de manufatura	48
4.4 - Aplicação do modelo hierárquico	50
5.1 - Escalonamento do <i>Job-shop</i> com paradas de produção e ociosidade	53
5.2 - Solução de escalonamento para o 3/3 JSSP	55
5.3 - Classificação das Técnicas utilizadas no JSSP	56
6.1 - Matriz com bloco na diagonal.....	57
6.2 - Algoritmo de Identificação de FP	57
6.3 - Matriz inicial para o Algoritmo CI.....	58
6.4 - Matriz após o passo um e dois do Algoritmo CI.....	58
6.5 - Matriz após o terceiro passo do Algoritmo CI.....	59
6.6 - Matriz após o quarto passo do Algoritmo CI.....	59
6.7 - (a) Matriz $k + 1$ (b) Matriz final $k + 1$ (c) Matriz $k + 2$	59
6.8 - Identificação da CM e FP a partir do algoritmo CI.....	60
6.9 - Estratégia de busca da BT	62
6.10 - Quatro principais dimensões da BT	63
6.11 - Algoritmo BT	63
7.1 - Ambiente do SMF	68
7.2 - (a) Robô (b) Esteira de transporte (c) Máquina (d) <i>Buffer</i> temporário	69
7.3 - Níveis de controle.....	70
7.4 - Arquitetura do modelo	73
8.1 - Diagrama de Classes	75
8.2 - Algoritmo para a geração de FP.....	77
8.3 - Escalonamento inicial após a terceira fase da arquitetura.....	79
8.4 - Escalonamento final após a quarta fase da arquitetura	81
8.5 - Algoritmo Busca Tabu para o modelo proposto	83
8.6 - (a) Lotes candidatos (b) Lotes retirados (as posições continuam).	85
8.7 - Seqüenciamento após a troca de dois lotes.	85
8.8 - Redefinição dos lotes nos turnos.....	86
8.9 - Movimento de troca de lotes no mesmo turno.	86
8.10 - Movimento de retirada da parte no lote origem.....	87
8.11 - Movimento de inserção da parte no lote destino.....	88

8.12 - Redefinição de lotes.	88
8.13 - Mínimos locais e ótimo global	89
8.14 - (a) Calendário fabril (b) seqüência das partes por máquina.....	90
9.1 - Ambiente de Manufatura para o estudo de caso.	92
9.2 - Características técnicas do produto alma de aço	94
9.3 - Gráfico solução real x solução inicial	97
9.4 - Gráfico solução real x não-tendenciosa.	99
9.5 - Gráfico solução real x não-tendenciosa x tendenciosa (variação p_1).....	101
9.6 - Comportamento das variáveis de decisão dada variação de p_1	102
9.7 - Gráfico solução real x não-tendenciosa x tendenciosa (variação p_2).....	104
9.8 - Comportamento das variáveis de decisão dada variação de p_2	105
9.9 - Gráfico solução real x não-tendenciosa x tendenciosa (variação p_3).....	107
9.10 - Comportamento das variáveis de decisão dada variação de p_3	107
9.11 - Gráfico solução real x não-tendenciosa x tendenciosa (variação p_4)	109
9.12 - Comportamento das variáveis de decisão dada variação de p_4	109
9.13 - Comportamento da variável atraso dado variação da lista tabu e o nbmax.....	113
9.14 - Estrutura genérica de uma instância do JSSP	113
9.15 - Gráfico <i>Gantt</i> para a instância FT06	116
10.1 - Gráfico comparativo instância LA16 (pequena escala)	123
10.2 - Gráfico comparativo instância LA40 (média escala).....	123

Lista de Tabelas

2.1 - Modelos de um sistema de manufatura	23
3.1 - Exemplo de estrutura hierárquica.....	30
3.2 - Matriz com fluxo parte x máquina	30
3.3 - Exemplo de codificação tipo cadeia para as máquinas	31
3.4 - Exemplo de codificação tipo cadeia para as partes	31
4.1 - Tipos de flexibilidade num SMF.....	44
8.1 - Informações de produção após a primeira fase da arquitetura	76
8.2 - Agrupamento das partes após a segunda fase da arquitetura	77
8.3 - Configuração após a aplicação da regra de despacho	79
8.4 - Seqüenciamento das partes na máquina após terceira fase da arquitetura	80
8.5 - Calendário fabril da máquina “515” após terceira fase da arquitetura.....	80
8.6 - Configuração após a aplicação do algoritmo BT	81
8.7 - Seqüenciamento das partes na máquina após quarta fase da arquitetura	82
8.8 - Calendário fabril da máquina “515” após quarta fase da arquitetura.....	82
8.9 - Plano de produção	90
9.1 - Paradas utilizadas no ambiente de manufatura	93
9.2 - Calendário fabril.....	94
9.3 - Solução inicial	96
9.4 - Solução obtida através do escalonamento real de produção	96
9.5 - Médias obtidas após os experimentos	98
9.6 - Solução não-tendenciosa	99
9.7 - Minimização do <i>makespan</i>	100
9.8 - Minimização do <i>makespan</i> desconsiderando o peso p_2	102
9.9 - Minimização do tempo total de atraso	103
9.10 - Minimização do tempo total parado.....	106
9.11 - Minimização do tempo total ocioso	108
9.12 - Valores utilizados para o $nbmax$ e a lista tabu	111
9.13 - Experimentos utilizando a estratégia A.....	111
9.14 - Experimentos utilizando a estratégia B.....	111
9.15 - Experimentos utilizando a estratégia C.....	112
9.16 - Experimentos utilizando a estratégia D.....	112
9.17 - Experimentos utilizando a estratégia E	112
9.18 - Autores selecionados para a validação do modelo.....	114
9.19 - Resultados dos experimentos	115
9.20 – Comparativo modelo proposto e o MSPF	117
10.1 – Resultados dos experimentos.....	122

Lista de Abreviaturas

AFP	- Análise por Fluxo de Produção
BEA	- <i>Bond Energy Algorithm</i>
BT	- Busca Tabu
CAD	- <i>Computer Aided Design</i>
CAM	- <i>Computer Aided Manufacturing</i>
CI	- <i>Cluster Identification</i>
CIM	- <i>Computer Integrated Manufacturing</i>
CM	- Células de Máquina
CN	- Controle Numérico
CNC	- <i>Computerized Numerical Control</i>
DNC	- <i>Distributed Numerical Control</i>
ECI	- <i>Extended Cluster Identification</i>
GRASP	- <i>Greedy Randomized Adaptive Procedure</i>
FP	- Família de Parte
JSSP	- <i>Job-shop Scheduling Problem</i>
NP	- <i>Non-Deterministic Polynomial-Time</i>
ROC	- <i>Rank Order Clustering</i>
SLCA	- <i>Single Linkage Cluster Analysis</i>
SM	- Sistemas de Manufatura
MSPF	- Modelo de Sequenciamento de Partes e de Ferramentas
SMF	- Sistema de Manufatura Flexível
TG	- Tecnologia de Grupo

Sumário

1	Introdução	15
2	Sistema de Manufatura	18
2.1	Definição	18
2.2	Componentes de um Sistema de Manufatura.....	19
2.2.1	Máquinas de produção	19
2.2.2	Sistema de Manuseio e Transporte de Materiais	19
2.2.3	Sistema de Controle Computacional.....	20
2.2.4	Recursos Humanos.....	21
2.3	Classificação de um Sistema de Manufatura.....	21
2.3.1	Tipos de Operações Executadas.....	21
2.3.2	Número de Estações de Trabalho e Sistemas de Leiaute	21
2.3.3	Níveis de Automação.....	21
2.3.4	Mix de Partes ou Produtos.....	22
2.4	Análise de Sistemas de Manufatura quanto a sua Classificação	23
3	Tecnologia de Grupo	25
3.1	Definições.....	26
3.2	Famílias de Partes.....	27
3.3	Benefícios da Implementação da TG.....	27
3.4	Métodos de Agrupamento.....	28
3.4.1	Inspeção Visual.....	29
3.4.2	Classificação por Codificação.....	29
3.4.2.a	Estrutura Hierárquica	29
3.4.2.b	Estrutura tipo Cadeia.....	30
3.4.2.c	Estrutura Mista	31
3.4.2.d	Seleção de um Sistema de Classificação e Codificação.....	31
3.4.2.e	Sistemas de Classificação e Codificação	32
3.4.2.e.1	Opitz.....	33
3.4.2.e.2	MULTICLASS.....	35
3.4.2.e.3	MICLASS	36
3.4.2.e.4	KK-3.....	36
3.4.3	Análise por Fluxo de Produção	37
3.4.3.a	Formulação Matricial	37
3.4.3.a.1	Medida de Desempenho.....	38
3.4.3.a.2	Trabalhos relacionados.....	40
4	Sistema de Manufatura Flexível.....	42
4.1	Definição	42
4.2	Flexibilidade	43
4.2.1	Qualificação	44
4.2.2	Número de Máquinas	45
4.3	Componentes	45
4.3.1	Máquinas.....	46
4.3.2	Estações de carregamento e descarregamento	46
4.3.3	Equipamentos para transporte.....	46
4.3.4	Paletes	47
4.3.5	Elementos de fixação	47
4.3.6	Ferramentas.....	47
4.3.7	Recursos humanos	47
4.3.8	Sistema computacional	47

4.4	Benefícios	48
4.5	Modelo conceitual	49
5	Problemas Abordados	51
5.1	Seleção de Partes	51
5.2	Escalonamento de Partes	51
5.2.1	O Problema do Escalonamento no Job-shop	52
5.2.1.a	Complexidade	53
5.2.1.b	Formulação Clássica	54
5.2.1.c	Exemplo de JSSP	55
5.2.1.d	Trabalhos relacionados	55
6	Técnicas utilizadas	57
6.1	Técnica para a Seleção de Partes	57
6.1.1	Algoritmo	57
6.1.2	Exemplo	58
6.1.3	Considerações	60
6.2	Técnica para o Escalonamento das Partes	60
6.2.1	Meta-heurística	60
6.2.2	Busca Tabu	62
6.2.3	Algoritmo	63
6.2.4	Aplicações	65
6.2.4.a	Escalonamento em Sistemas de Manufatura	66
6.2.4.b	O problema do Roteamento de Veículos	66
6.2.4.c	Modelo de escalonamento de partes em uma máquina versátil	67
6.2.4.d	Auditoria de Escalonamento de produção	67
7	Modelo Proposto	68
7.1	Ambiente do Sistema de Manufatura Flexível	68
7.2	Formulação do Modelo	70
7.3	Arquitetura do Modelo	73
8	Aplicação	75
8.1.1	Demanda de Produção	76
8.1.2	Geração das Famílias de Partes	76
8.1.3	Geração da Solução Inicial	78
8.1.4	Geração do Escalonamento	80
8.1.4.a	Algoritmo Busca Tabu	82
8.1.4.a.1	Geração de Vizinhança	84
8.1.4.a.2	Movimento de troca de lotes	84
8.1.4.a.3	Movimento de retirada e inserção de parte	87
8.1.4.a.4	Lista Tabu	88
8.1.4.a.5	Número de iterações	89
8.1.5	Gravação do Plano de Produção	89
9	Experimentos e Validação	91
9.1	Experimento	91
9.1.1	Estudo de caso	92
9.1.1.a	O ambiente	92
9.1.1.b	O produto	94
9.1.2	Parâmetros considerados no modelo	95
9.1.3	Solução inicial	96
9.1.4	Solução tendenciosa e não-tendenciosa	98
9.1.5	Otimização do Tempo Total de Produção	100
9.1.6	Otimização do Tempo Total de Atraso	103

9.1.7	Otimização do Tempo Total Parado	105
9.1.8	Otimização do Tempo Total Ocioso	108
9.1.9	Lista Tabu e o nbmax.....	110
9.2	Validação	113
9.2.1	Trabalhos relacionados	114
9.2.2	Configuração da validação.....	114
9.2.3	Resultados computacionais.....	115
9.2.4	Comparativo com o Modelo de Seqüenciamento de Partes e de Ferramentas.....	117
10	Conclusões	119
10.1	Problema da Seleção de Partes e do Escalonamento	119
10.2	Resultados dos experimentos e validação	120
10.3	Características do Ambiente de Manufatura	123
10.4	Trabalhos futuros	124
	Referências Bibliográficas	125

1 Introdução

Existem problemas que demandam um grande esforço matemático para sua resolução, devido à complexidade do algoritmo, a estrutura de dados utilizada, aos recursos computacionais disponíveis e, principalmente, em função da dimensão da entrada de dados. Para esses problemas, nem sempre é possível garantir que o resultado obtido represente a solução ótima. Nestes casos, a utilização de diferentes técnicas pode obter soluções com diferentes graus de qualidade. Busca-se então, uma técnica, que gere uma solução com melhor qualidade que as outras já obtidas e que atenda ao problema proposto.

Inúmeras técnicas de otimização foram desenvolvidas para a solução de problemas de otimização combinatória, destacando-se, problemas de representação (Gráfico de *Gantt* e Disjunção de Grafos), algoritmos de otimização (métodos eficientes de solução em tempo polinomial, programação matemática, *branch and bound*) e algoritmos de aproximação (métodos construtivos por regras de despacho por prioridades, métodos construtivos por heurísticas baseadas em gargalos, métodos construtivos por algoritmos de Inserção, busca local e métodos iterativos).

Muitas dessas técnicas produzem resultados considerados satisfatórios, porém, não há garantias quanto à existência de uma solução melhor do que aquela encontrada em um tempo computacional aceitável. Segundo Garey [Garey, 1976], o problema de otimização combinatória é uma tarefa de difícil complexidade e é conhecido na teoria da complexidade como NP-Difícil [Papadimitriou, 1995].

Podemos considerar como exemplo, para o problema supracitado, a minimização do tempo total de produção, através do escalonamento de um conjunto de “n” *jobs* a serem processados num conjunto de “m” máquinas. Cada *job* é composto de “i” operações que devem ser processadas através de uma rota; para cada operação é definido uma máquina e tempo de processamento padrão.

A inexistência de uma solução melhor é consequência da limitação dessas técnicas em realizar uma determinada pesquisa, em tempo hábil, em todo o espaço amostral do problema em questão. Para contornar essa limitação, as técnicas desenvolvidas até o momento propõem a redução desse espaço através de regiões, a fim de obter resultados ótimos destas regiões visando o ótimo global do problema em questão [Viana, 1998].

Diversas áreas como a de telecomunicações, logística, metal-mecânico e componentes para calçados apresentam problemas com as características mencionadas anteriormente, ou seja, de otimização combinatória (exemplo: escalonamento de partes). Portanto, são atividades que tem fomentado o desenvolvimento de técnicas de otimização eficientes que sejam capazes de tratar, de forma adequada, a complexidade dos problemas de otimização atuais.

No entanto, estas áreas, na busca da excelência no mercado através do investimento tecnológico, da utilização eficiente dos recursos, na automação de máquinas/processos e na qualificação de recursos humanos, objetivam o principal ganho estratégico que é o aumento da produtividade. A busca da produtividade conflita com outra característica de excelência do mercado para uma indústria que é a flexibilidade do Sistema de Manufatura, ou seja, a capacidade de reprogramar a produção através de um *mix* de diferentes estilos de partes, respeitar as datas de entrega do cliente, o aproveitamento integral das máquinas de produção, a facilidade de processar uma nova parte tão logo que esta esteja com as características do projeto definidas, a redução do trabalho em processo e a facilidade na adaptação do sistema a quaisquer mudanças no mercado.

Gerenciar o conflito entre a flexibilidade, característica de um Sistema de Manufatura Flexível, e a produtividade, característica de um sistema de produção em massa (*Flow-shop*), são fatores que motivam este trabalho.

Portanto, esta dissertação tem como objetivo a geração de um modelo de escalonamento aplicado ao *Job-shop Scheduling Problem* (JSSP) considerando um Sistema de Manufatura Flexível (SMF), visando otimizar o tempo total de produção (*makespan*), o tempo total de atraso, o tempo total de parada e o tempo total ocioso dos turnos.

Para tanto, a dissertação está organizada da seguinte forma. No capítulo 2 é apresentada uma introdução aos Sistemas de Manufatura, caracterizando seus estágios de evolução. No capítulo 3 é contextualizada a Tecnologia de Grupo, enfatizando sua aplicabilidade, aspectos históricos, técnicas e algoritmos para a geração de Famílias de Partes.

A introdução aos Sistemas de Manufatura Flexível, apresentando seus componentes, aplicações, benefícios, o modelo conceitual e o enfoque à flexibilidade são abordados no capítulo 4. No capítulo 5 são abordados os problemas da Seleção de Partes e o problema de Escalonamento bem como um estudo do JSSP, enfatizando sua formulação clássica, complexidade do problema e o conflito entre a flexibilidade versus produtividade.

No capítulo 6 é apresentada a técnica de agrupamento de partes, para a solução do problema da Seleção de Partes e, para o problema do Escalonamento de Partes, é apresentada a técnica Busca Tabu.

O desenvolvimento do modelo aplicado ao JSSP através do ambiente de manufatura, da formulação e arquitetura é apresentado no capítulo 7. A aplicação do modelo, detalhando a entrada, processamento e saída de cada fase da arquitetura, é apresentada no capítulo 8. Os experimentos realizados a partir de um estudo de caso e a validação do modelo através de trabalhos clássicos de JSSP são mostrados no capítulo 9. Finalmente, no capítulo 10 são apresentadas as conclusões e trabalhos futuros referentes a esta dissertação, e no capítulo 11, as referências bibliográficas.

2 Sistema de Manufatura

Em uma fábrica, as facilidades de produção estão relacionadas às máquinas, ferramentas, equipamentos de manuseio e transporte de material, equipamentos de inspeção e sistemas computacionais para controlar as operações de produção. Tais recursos são arranjados em grupos lógicos de organização formando um Sistema de Manufatura [Tempelmeier, 1993; Wu, 1994]. A seguir são apresentadas as definições, componentes e classificação dos SMF que por sua vez, darão suporte na descrição do modelo que será desenvolvido.

2.1 Definição

Um Sistema de Manufatura (SM) é definido como uma coleção de equipamentos e recursos humanos, integrados de maneira a fornecer uma ou mais operações de produção, iniciando da matéria-prima ou de uma parte ou conjunto de partes. Estes equipamentos integrados incluem máquinas, ferramentas, sistema de manuseio de material, dispositivos de posicionamento, recursos humanos e sistemas computacionais [Jha 1991; Tempelmeier, 1993; Wu, 1994; Groover, 2001]. A figura abaixo contextualiza o Sistema de Manufatura em um Sistema de Produção.

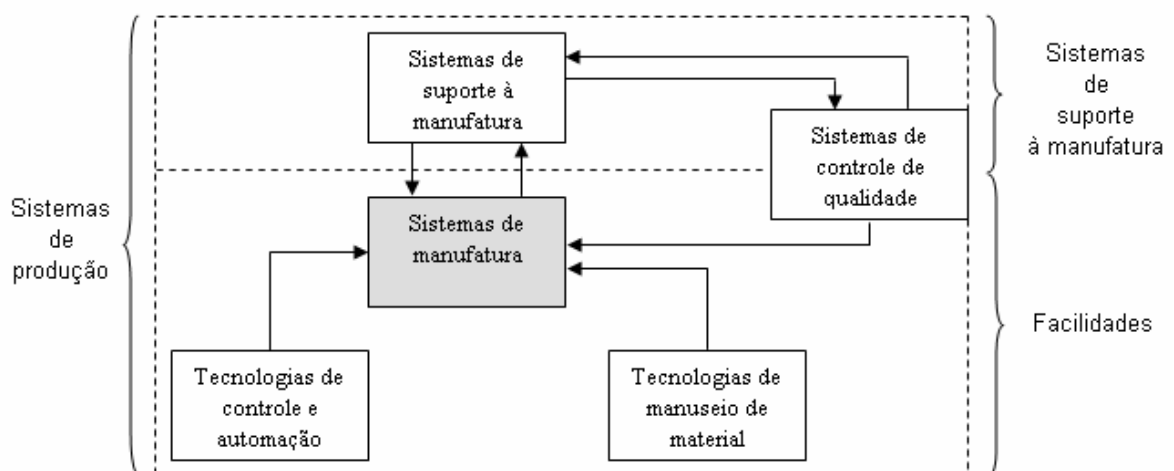


Figura 2.1 - Posicionamento dos SM em um Sistema de Produção [Groover, 2001].

Pode-se citar como exemplos de Sistema de Manufatura, um trabalhador operando uma máquina com operações semi-automatizadas, um grupo de máquinas semi-automatizadas operadas por um trabalhador, uma máquina de montagem automatizada sendo esta operada periodicamente por um trabalhador, um grupo de máquinas automatizadas e um grupo de trabalhadores realizando operações de montagem numa linha de produção [Wu, 1994; Groover, 2001].

2.2 Componentes de um Sistema de Manufatura

Existem diversos componentes em um Sistema de Manufatura. Contudo, normalmente estes componentes sempre se relacionam às máquinas de produção, ao sistema de manuseio de material, os sistemas computacionais e aos recursos humanos [Lorini, 1993].

2.2.1 Máquinas de produção

As máquinas são classificadas quanto a sua operacionalidade, ou seja, o grau de automação [Jha, 1991]. Existem três grupos distintos:

- operação manual: são máquinas controladas ou supervisionadas por um trabalhador. As máquinas fornecem a força para a operação e os trabalhadores realizam o controle. Exemplo: torno mecânico e uma injetora de solado;
- operação semi-automatizada: são máquinas que realizam uma parte do ciclo de trabalho sobre um programa de controle, e trabalhadores realizam o controle da outra parte do ciclo. Exemplo: máquinas CNC; e
- operação automatizada: são máquinas que possuem a capacidade de operar sem intervenção de trabalhadores por períodos longos de ciclos. Exemplo: Injeção de moldes, onde a máquina executa o ciclo automatizado, mas periodicamente o repositório estando completo, este deve ser substituído.

2.2.2 Sistema de Manuseio e Transporte de Materiais

Um Sistema de Manuseio e Transporte de Materiais deve ter as seguintes características [Lorini, 1993]: carregamento e descarregamento de material, posicionamento de material em uma unidade de trabalho, atuar como unidade de transporte do material entre estações, sendo que o roteamento pode ser fixo ou variado e seu carregamento manual ou por equipamento de transporte apropriado e realizar a função de armazenamento temporário.

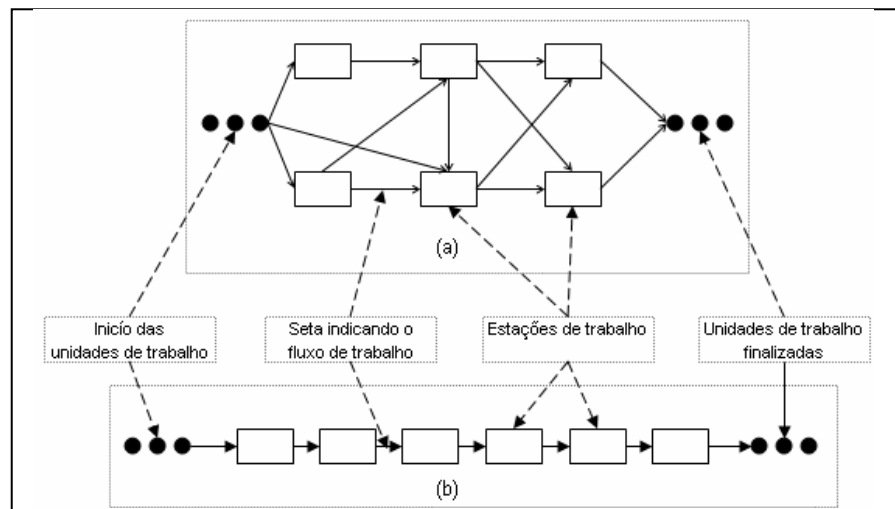


Figura 2.2 - (a) Roteamento variado e (b) Roteamento fixo [Groover, 2001]

2.2.3 Sistema de Controle Computacional

Num Sistema de Manufatura, um sistema computacional é requisito para realizar o controle tanto em equipamentos automatizados como semi-automatizados, sendo este essencial para o controle e gerenciamento da manufatura. Algumas funcionalidades dos sistemas computacionais são apresentadas a seguir [Wu, 1994; Groover, 2001]:

- instruções de comunicação para os trabalhadores: em estações de trabalho operadas manualmente que executam diferentes tarefas sobre diferentes unidades de trabalho, instruções de processamento e montagem para uma unidade específica devem ser comunicadas para o trabalhador;
- *download* de programas de partes para DNC;
- sistema de manuseio e transporte de materiais: controlar as coordenadas e atividades das estações de trabalho;
- programação da produção;
- diagnóstico de falhas;
- monitoramento de segurança;
- controle de qualidade; e
- gerenciamento das operações.

2.2.4 Recursos Humanos

Pessoas são essenciais para o controle, gerenciamento, manutenção e suporte dos Sistemas de Manufatura. Em sistemas automatizados, os recursos humanos podem ser utilizados em tarefas de carregamento e descarregamentos de partes entre sistemas, em trocas de ferramentas e outras funções similares [Lorini, 1993].

2.3 Classificação de um Sistema de Manufatura

Através de uma variedade de tipos de Sistemas de Manufatura, a classificação destes originaram-se de alguns fatores como, tipos de operações executadas, número de estações de trabalho e sistemas de leiaute, nível de automação e *mix* de partes ou produtos [Tempelmeier, 1993; Wu, 1994].

2.3.1 Tipos de Operações Executadas

Primeiramente, todo Sistema de Manufatura é distinguido pelos tipos de operações que eles executam. O nível mais alto está na distinção entre operações de processamento sobre unidades de trabalho e operações de montagem combinando partes individuais em conjuntos. Além destas distinções, existem tecnologias de processamento individual e operações de montagem. Exemplo: (a) curtimento de couro e (b) montagem de carro [Groover, 2001].

2.3.2 Número de Estações de Trabalho e Sistemas de Leiaute

O número de estações de trabalho é o fator chave para este esquema de classificação, pois influenciam na capacidade de produção, produtividade, custo por unidade e manutenção. Existem três níveis de classificação, quanto ao número de estações de trabalho e sistemas de leiaute, sendo elas: estação única (tipo I), estações múltiplas com roteamento variável (tipo II) e estações múltiplas com roteamento fixo (tipo III) [Kusiak, 1994; Groover, 2001].

2.3.3 Níveis de Automação

O nível de automação é outro fator que caracteriza um Sistema de Manufatura. Pode ser manual, semi-automatizada (híbrida) e automatizada [Groover, 2001]. Exemplo:

- única estação manual (IM);
- única estação automatizada (IA);

- multi-estação com sistema manual e roteamento variado (IIM);
- multi-estação com sistema automatizado e roteamento variado (IIA);
- multi-estação com sistema híbrido e roteamento variado (IIH);
- multi-estação com sistema manual e roteamento fixo (IIIM);
- multi-estação com sistema automatizado e roteamento fixo (IIIA); e
- multi-estação com sistema híbrido e roteamento fixo (IIIH).

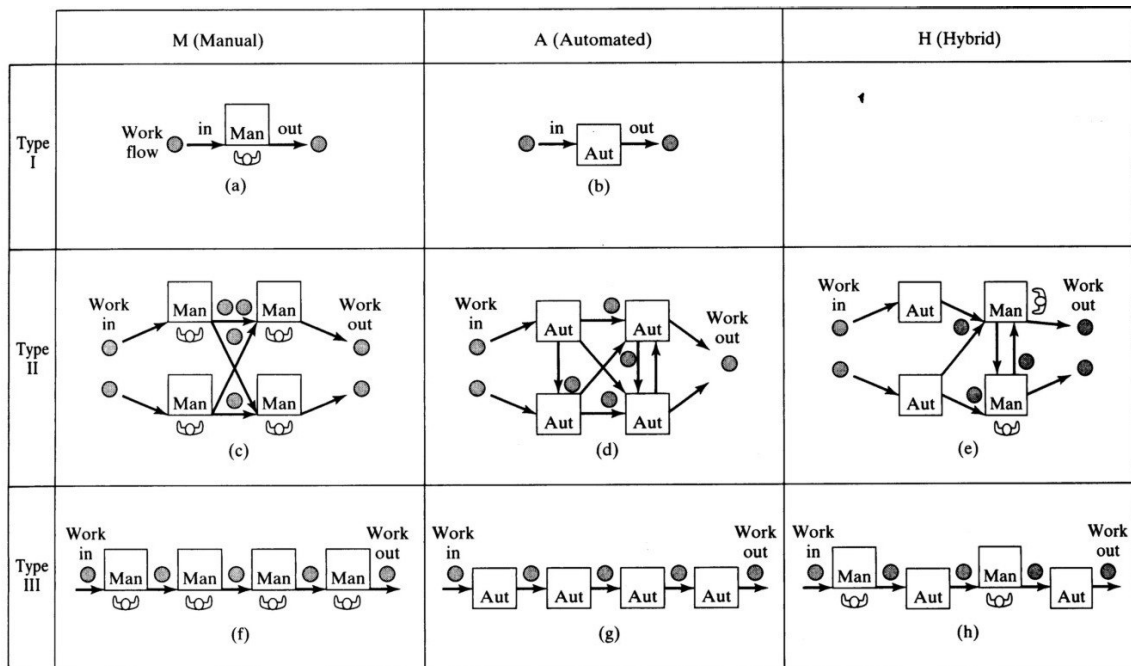


Figura 2.3 - Classificação dos SM quanto ao Nível de Automação [Groover, 2001].

2.3.4 Mix de Partes ou Produtos

A última característica dos Sistemas de Manufatura está relacionada com a capacidade de produção em relação ao *mix* de estilos de partes ou produtos a serem produzidos. A tabela abaixo apresenta três modelos de Sistemas de Manufatura de acordo com a capacidade em gerenciar o *mix* de produtos [Kusiak, 1992].

Tabela 2.1 – Modelos de um sistema de manufatura

Modelo	Característica
Simple	Todas as partes ou produtos são feitos por um sistema de manufatura idêntico sem variações. Neste caso a demanda para o item deve ser suficiente para justificar dedicação do sistema de produção por um período extenso.
Lote (<i>batch</i>)	Diferentes partes ou produtos são feitos pelo sistema, porém, a cada mudança de grupo é necessário uma mudança de <i>setup</i> ou equipamento programado.
Mix	Diferentes partes ou produtos são feitos pelo sistema, porém o sistema está apto para manusear estas diferenças sem necessitar mudança de <i>setup</i> ou equipamento programado. Podemos destacar neste modelo o Sistema de Manufatura Flexível (SMF)

Fonte: Groover [Groover, 2001].

2.4 Análise de Sistemas de Manufatura quanto a sua Classificação

Após a definição da classificação dos Sistemas de Manufatura, é possível traçar dois gráficos que são apresentados na Figura 2.4.

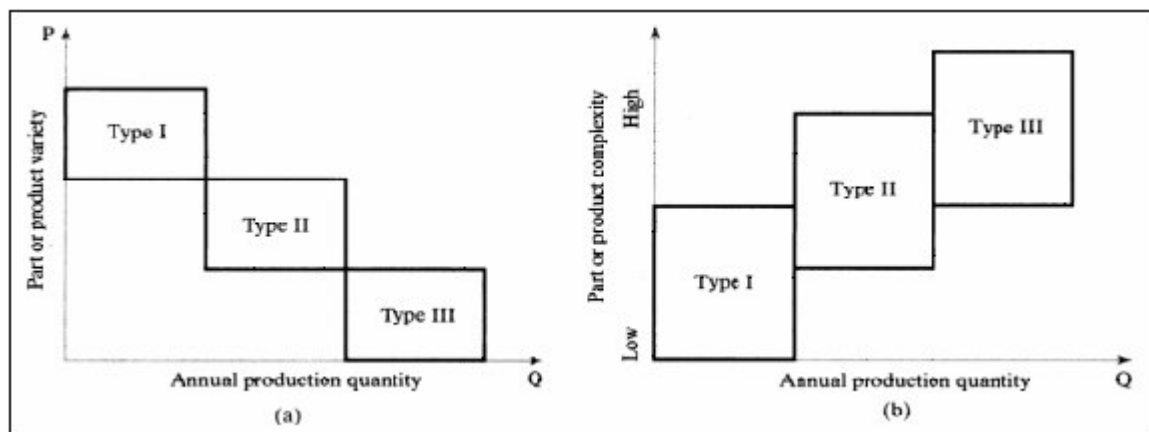


Figura 2.4 - (a) Flexibilidade e Produtividade e (b) Complexidade nos SM [Groover, 2001].

A Figura 2.4a apresenta a relação entre Flexibilidade e a Produtividade de vários tipos de Sistemas de Manufatura quanto ao grau de automação. Os sistemas tipo I (sistemas manuais) possuem maior Flexibilidade em virtude do trabalho artesanal, possibilitando assim trabalhar com *mix* de estilos de partes ou produtos. Contudo, quanto mais Flexibilidade o sistema possui, menor é a quantidade de produção resultante desse sistema, em virtude da

baixa Produtividade. Os sistemas tipo III possuem quantidade de produção elevada, porém a variedade de partes é menor, resultando assim na produção em massa (*flow-shop*).

Já na Figura 2.4b são ilustradas a relação do grau de complexidade de partes ou produtos e a Produtividade. Se um produto tem baixa complexidade de produção e baixo volume de produção então não se justifica ter um SM tipo II ou tipo III. É mais adequada a utilização de um SM tipo I. Quanto mais complexo é o estilo de parte ou produto, os sistemas do tipo III (sistemas automatizados) são mais apropriados, permitindo assim um volume alto de produção, ou seja, sistemas de produção em massa. No momento em que a complexidade é baixa, os sistemas do tipo II (sistemas semi-automatizados) são mais apropriados. No entanto, métodos da Tecnologia de Grupo serão apresentados no capítulo 3, na qual permite gerenciar o conflito entre a flexibilidade e a produtividade.

3 Tecnologia de Grupo

Neste capítulo são apresentados definições, benefícios e técnicas de agrupamento da Tecnologia de Grupo (TG) que serão utilizadas no método de agrupamento das partes para o modelo proposto.

A TG é bastante recente, porém o pensamento em produzir partes com características semelhantes de maneira agrupada utilizando métodos de inspeção visual ou até mesmo por análise de fluxo em uma linha de produção são bem antigos. Atualmente, a diferença principal está na utilização de técnicas, conceitos organizados e recursos modernos [Lorini, 1993].

Embora tenham sido publicados alguns trabalhos nas décadas de 20 a 40 enfocando o agrupamento de máquinas e produção em lotes, é na década de 50 que a idéia de tecnologia de grupo começa a se tornar consistente [Lorini, 1993]. Com base em Irani e Groover [Irani, 1999; Groover, 2001] é apresentado uma evolução do estado da arte referente a TG:

- em 1925, *R. Flanders* apresentou um artigo nos Estados Unidos no qual descrevia uma maneira de organizar a manufatura. A empresa Lamson e Jones chamou o método de organização de Tecnologia de Grupo;
- em 1937, *A. Sokolovsky* da União Soviética descreveu as características essenciais da Tecnologia de Grupo propondo um agrupamento de partes similares para serem produzidas em seqüências de processos padrões, permitindo assim, técnicas de linhas de produção (processos de lotes);
- em 1949, *A. Korling* na *Scania-Vabis* apresentou um artigo na França com o título “Produção de Grupo”. Diversos princípios na adaptação das técnicas nas linhas de produção para produção em lotes;
- em 1959, o pesquisador *S. Mitrofanov* da União Soviética publicou um livro com o título *Scientific Principles of Group Technology*. *Mitrofanov* é responsável por mais de 800 plantas usando tecnologia de grupo (até 1965). Outro pesquisador, *H. Opitz* na Alemanha estudou a classificação e sistemas de codificação para as partes. Posteriormente foi gerado um sistema de classificação com o seu nome *Opitz*;
- em 1962 na Itália, *Burbidge* publicou o método de análise de fluxo de produção sendo também utilizado para a formação de famílias em Tecnologia de Grupo;
- a partir de 1965 ocorreu a criação de centros de pesquisa em diversos países;

- a partir de 1967 a Sociedade Japonesa para Promoção de Máquinas (JSPMI) promoveu importantes estudos sobre a Tecnologia de Grupo. As pesquisas foram realizadas em cooperação com o Laboratório de Engenharia Mecânica (*Kikai Gijutsu Kenkyujo*), Centro Geral de Fundação (*So-go Iomo Center*), empresas e universidades. Alguns sistemas de classificação dentre os mais importantes podem ser destacados: o KK-1 (1970), o KK-2 (1973) e o KK-3 (1976). Desde 1960 os Japoneses vem utilizando a TG juntamente com outras técnicas como o *Just-in-Time* obtendo excelentes resultados [Lorini, 1993];
- nos Estados Unidos, a primeira aplicação de Tecnologia de Grupos, foi através da *Langston Division of Harris-Intertype* em *New Jersey* por volta de 1969. A empresa aplicou o conceito de famílias de partes atingindo 50% no aumento da produtividade e o tempo de atraso antes em semanas, agora era em dias; e
- as tentativas de utilização da TG no Brasil ainda são tímidas em relação ao potencial produtivo que a TG pode fornecer [Lorini, 1993].

3.1 Definições

Para Kusiak e Irani [Kusiak, 1992; Irani 1999], a Tecnologia de Grupo é uma filosofia, na qual as partes são identificadas e agrupadas segundo similaridades, com o objetivo de obter maior integração entre o projeto e a manufatura e aumentar a produtividade. O agrupamento de partes por similaridades é denominado de Família de Partes [Kusiak, 1992; Lorini, 1993; Bedworth, 1991], onde cada Família possui características similares de projeto e ou manufatura. A utilidade do conceito das Famílias de Partes pode ser comprovada, através do seguinte exemplo: Numa produção de 10.000 diferentes partes pode-se atingir aproximadamente 30 a 40 famílias distintas. Dessa forma é mais fácil trabalhar com 30 a 40 Famílias de Partes do que com 10.000 partes distintas.

Segundo Lorini [Lorini, 1993], a Tecnologia de Grupo é uma filosofia que explora as semelhanças para se obter vantagem operacional e econômica através de um tratamento de grupo.

Para Kusiak, Slack e Groover [Kusiak, 1992; Slack, 1997; Groover, 2001], a constatação de muitos problemas de produção são similares, e que o agrupamento destes problemas pode levar a obtenção de uma única solução que resolva o agrupamento, certamente poupará tempo e esforço.

3.2 Famílias de Partes

São uma coleção de partes que são similares pela sua forma geométrica ou em virtude das etapas de processo similares necessárias a sua produção. Os tipos de similaridades são provenientes das características de projetos (formas geométricas e suas dimensões) ou características da Manufatura (tipos e seqüenciamentos de processos). A migração para a TG está justamente na dificuldade da geração das Famílias de Partes (FP). As Figuras 3.1 e 3.2 ilustram as FP [Bedworth, 1991; Kusiak, 1992; Lorini, 1993].

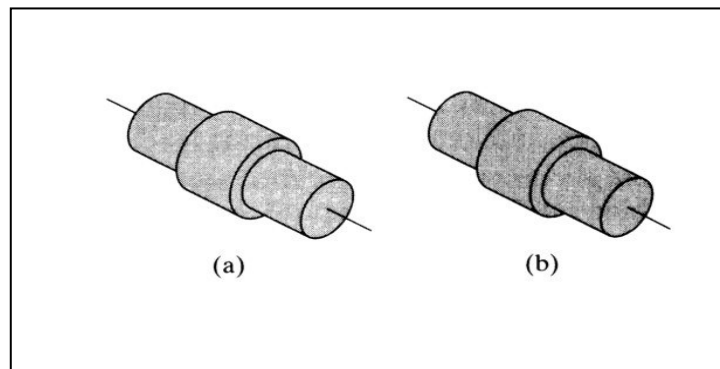


Figura 3.1 - Partes com durezas diferentes [Groover, 2001]

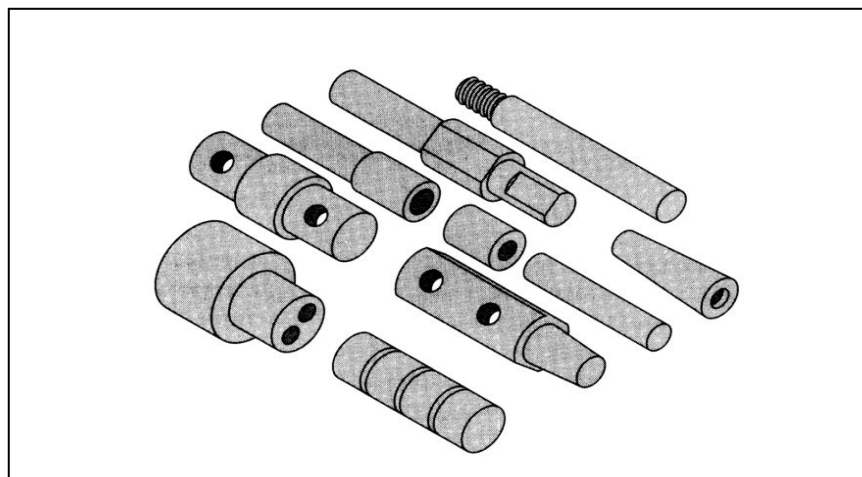


Figura 3.2 - Partes com processos similares e atributos diferentes [Groover, 2001].

3.3 Benefícios da Implementação da TG

A TG oferece benefícios substanciais para as empresas que realizam a implantação. Alguns benefícios são apresentados abaixo [Groover, 2001]:

- promove a padronização de ferramentas, instalações e *setup*;

- reduz o manuseio do material porque as partes são movidas somente nas células de máquinas e não pela fábrica;
- o planejamento do processo e a programação da produção são simplificados;
- os tempos de *setup* são reduzidos, resultando em diminuição de atrasos na produção;
- o trabalho em processo é reduzido;
- o comprometimento e satisfação dos trabalhadores melhoram numa célula com TG;
- é realizado um trabalho de alta qualidade quando utilizada a TG;
- redução do custo de material em processo de estoque, ou seja, ocorre a redução do ciclo de fabricação melhorando a resposta da demanda e a diminuição dos estoques intermediários combinados ao estoque final; e
- com o uso do CAD/CAM, manufatura em célula e CIM, obtêm-se resultados significativos na produtividade e na diminuição de custos na fabricação de lotes pequenos. Dependendo do nível de implementação, potencialmente podem-se atingir reduções de 5% a 75%.

Segundo Kusiak [Kusiak, 1990], a TG oferece as seguintes vantagens:

- redução de atraso (20% a 88%);
- redução do trabalho em processo (acima de 88%);
- redução do trabalho (15% a 25%);
- redução do uso de ferramentas (20% a 30%);
- redução de retrabalho e de perdas (15% a 75%);
- redução de tempo de *setup* (20% a 60%);
- redução no tempo de entrega (13% a 36%);
- melhoria nas relações humanas; e
- redução de documentos.

3.4 Métodos de Agrupamento

A grande dificuldade em implementar a TG, em um Sistema de Manufatura, é a geração de FP [Lorini, 1993]. Para tanto, são citados três métodos para a geração de FP, sendo eles: inspeção visual, classificação por codificação e a análise por fluxo de produção (AFP) [Jha, 1991].

3.4.1 Inspeção Visual

O método de Inspeção Visual é um procedimento semi-sistêmico e pouco sofisticado, ou seja, as partes são agrupadas de acordo com as similaridades geométricas visualmente. O agrupamento das partes neste método depende do fator humano para obter sucesso, podendo causar inconsistências. É aplicado nos casos onde são poucos os tipos de partes [Kusiak, 1994].

3.4.2 Classificação por Codificação

Neste método, as partes podem ser classificadas através das seguintes características: forma geométrica e complexidade, dimensões, tipo de material, matéria-prima e requisitos de medidas na parte final [Jha, 1991].

Lorini [Lorini, 1993] apresenta os motivos para a utilização dos esquemas de codificação, sendo eles:

- recuperação de projeto: o projetista tendo em mãos características do projeto, pode simplesmente identificá-lo e a partir deste projeto existente, alterar somente as características inerentes do novo projeto (demandaria menos tempo);
- planejamento de processos automatizados: a codificação de uma nova parte pode ser usada para procurar em planos de processos a existência de partes com código idêntico ou similar; e
- projeto de uma célula de máquina: os códigos das partes podem ser utilizados para projetar células de máquinas capazes de produzir todos os membros daquela família de parte particular, usando a composição da parte criada.

Para Jha [Jha, 1991] existem três estruturas na classificação por codificação, que por sua vez baseiam-se nos atributos do projeto ou nos atributos da Manufatura ou em ambos: a estrutura hierárquica (Tabela 3.1), a estrutura tipo cadeia (Tabela 3.2) e a estrutura mista.

3.4.2.a Estrutura Hierárquica

Cada símbolo sucessivo da codificação depende do valor do símbolo predecessor. A Tabela 3.1 ilustra a estrutura hierárquica.

Tabela 3.1 – Exemplo de estrutura hierárquica

Codificação	Descrição
1	Aço
1.1	Inox
1.1.1	Cromo
1.1.2	Espessura
2	Plástico
2.1	PU
2.1.1	PU-1060
2.1.2	PU-1070U
2.2	TR

É apresentada a codificação “1.1.2” para o item “Espessura”. Neste caso, a espessura é para o Aço Inox, onde o item “Inox” possui o código estruturado “1.1” e o “Aço” possui código estruturado “1”, ilustrando assim a hierarquia da estrutura.

3.4.2.b Estrutura tipo Cadeia

Cada símbolo na seqüência é sempre o mesmo, ou seja, não depende do valor dos símbolos predecessores.

Tabela 3.2 – Matriz com fluxo parte x máquina

Máquina	Parte													
	a	b	c	d	e	f	g	h	i	j	k	l	m	n
A-B														
C								x	x		x		x	x
D								x	x	x	x			
E-F		x		x			x							
G	x					x						x		
H-I	x	x	x	x	x	x						x	x	x
J-K	x	x	x	x	x	x						x	x	x

Fonte: Lorini [Lorini, 1993].

A Tabela 3.2 apresenta o fluxo de produção para cada parte gerando assim a matriz parte versus máquina.

Tabela 3.3 – Exemplo de codificação tipo cadeia para as máquinas

Máquina	Codificação
A-B (furadeira)	1
C (torno)	2
D (prensa)	3
E-F (fresadora)	4
G (politriz)	5
H-I (retífica)	6
J-K (serra)	7

Fonte: Lorini [Lorini, 1003].

Após a matriz de fluxos, é gerada a codificação em função das máquinas utilizadas pelas partes. A Tabela 3.3 apresenta a codificação tipo cadeia para as máquinas.

Tabela 3.4 – Exemplo de codificação tipo cadeia para as partes

Parte	Processo
a, f, l	567
c, e	467
b, d, g	367
m, n	167
h, k	12
j	24
i	124

A Tabela 3.4 ilustra o esquema de codificação tipo cadeia para as partes. A parte “a” será composta da codificação “567”, ou seja, utiliza as máquinas “G”, “H-I” e “J-K”.

3.4.2.c Estrutura Mista

É composto pela codificação da estrutura hierárquica e pela estrutura tipo cadeia. Exemplo: a parte “a” possui o seguinte código: 212.567, onde “212” é o material PU-1070U obtido da estrutura hierárquica (Tabela 3.1) e o código “567” da codificação tipo cadeia resultante da tabela de partes (Tabela 3.4).

3.4.2.d Seleção de um Sistema de Classificação e Codificação

Em virtude das particularidades encontradas nas empresas, não existe um sistema de codificação que seja utilizado em sua plenitude, ou seja, atenda às particularidades do produto

[Lorini, 1993]. Através desta concepção, alguns aspectos devem ser considerados para que aconteça uma escolha adequada, sendo elas:

- objetivo: os objetivos principais da aplicação, quanto a sua aplicabilidade e atributos devem estar bem definidos para que ocorra uma escolha coerente;
- abrangência: o sistema deve atender todas as partes ou produtos da empresa;
- expansibilidade: é fundamental definir um sistema em que a sua capacidade de expandir seja flexível, justamente para atender situações futuras;
- automação: através de um suporte computacional deve-se obter a integração e afinidade com demais sistemas da empresa;
- eficiência: a utilização de todos os atributos do sistema para a parte ou produto, sem que ocorram níveis desnecessários;
- simplicidade: o sistema deve ser simples para que usuários da aplicação possam gerenciar o sistema; e
- custos: os custos podem abranger vários aspectos, sejam custos de implantação ou de manutenção.

3.4.2.e Sistemas de Classificação e Codificação

Em Bedworth, Askin, Lorini e Groover [Bedworth, 1991; Askin, 1993; Lorini, 1993; Groover, 2001] são apresentados alguns dos principais sistemas de classificação e codificação, sendo eles: (a) sistema de classificação Opitz, (b) o sistema *Brisch* da *Brisch-Birn Inc*, (c) o sistema CODE desenvolvida pela *Manufacturing Data Systems*, (d) o CUTPLAN da *MetCut Associates*, (e) o DCLASS da *Brigham Young University*, (f) o sistema de classificação MultiClass desenvolvido pela *Organization for Industrial Research*, (g) o *Part Analog System* desenvolvido pela *Lovelace, Lawrence & Co*, (h) o sistema MICLASS da *Netherlands Organization for Applied Scientific Research* e (i) o sistema KK-3 da *Japan Society for the Promotion of Machine Industry*.

Pode-se destacar o sistema de Classificação brasileiro chamado (j) SCC/GRUCON em que a proposta desta codificação está em atender indústrias do ramo metal-mecânico [Lorini, 1993]. A seguir, são demonstrados os esquemas de classificação e codificação dos sistemas Opitz, MULTICLASS, MICLASS e o KK-3.

3.4.2.e.1 Opitz

O modelo Opitz foi desenvolvido na Alemanha em 1960 por *H. Opitz*. O código básico consiste de nove dígitos que representam dados de projeto e de fabricação. Quatro códigos adicionais podem ser utilizados para identificar o tipo e a seqüência das operações de fabricação [Askin, 1993]. Existem dois problemas relacionados a esta codificação: podem-se obter códigos diferentes para partes que possuam atributos de fabricação similares e partes com formas diferentes podem ter o mesmo código.

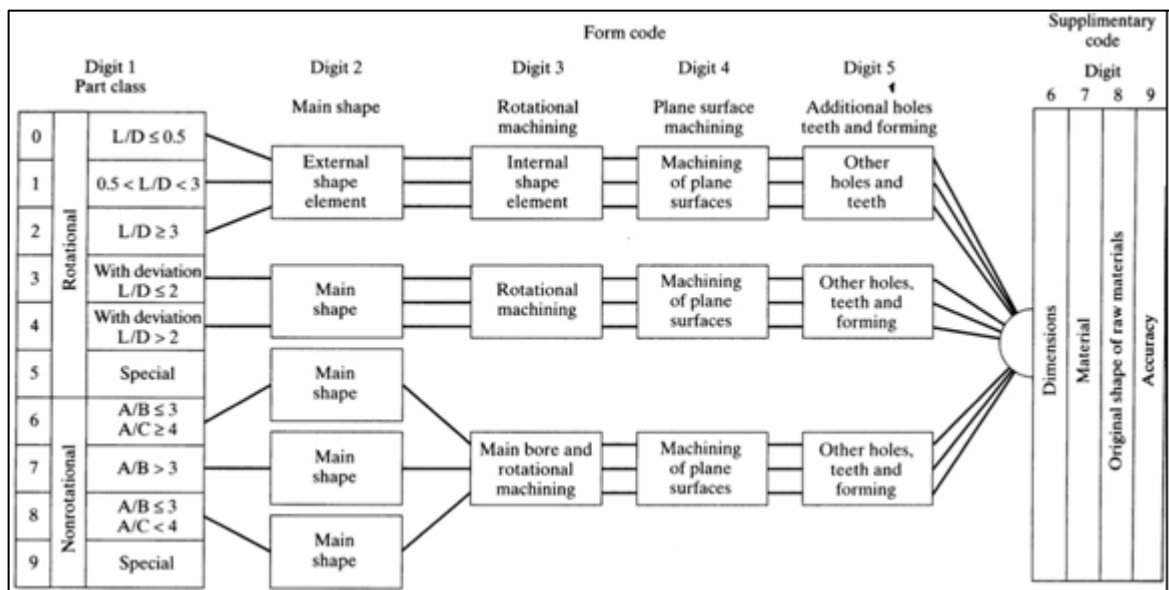


Figura 3.3 - Esquema de codificação do modelo Opitz [Askin, 1993].

Digit 1		Digit 2		Digit 3		Digit 4		Digit 5													
Part class		External shape, external shape elements		Internal shape, internal shape elements		Plane surface machining		Auxiliary holes and gear teeth													
Rotational parts	0	L/D ≤ 0.5		0		Smooth, no shape elements		0		No hole, no breakthrough		0		No surface machining		0		No auxiliary hole			
	1	0.5 < L/D < 3		1		No shape elements		1		No shape elements		1		Surface plane and/or curved in one direction, external		1		Axial, not on pitch circle diameter			
	2	L/D ≥ 3		2		Thread		2		Thread		2		External plane surface related by graduation around the circle		2		Axial on pitch circle diameter			
	3			3		Stepped to one end or smooth		Functional groove		3		Functional groove		3		External groove and/or slot		3		Radial, not on pitch circle diameter	
	4			4		Stepped to both ends		No shape elements		4		No shape elements		4		External spline (polygon)		4		Axial and/or radial and/or other direction	
Nonrotational parts	5			5		Stepped to both ends		Thread		5		Thread		5		External plane surface and/or slot, external spline		5		Axial and/or radial on PCD and/or other directions	
	6			6		Stepped to both ends		Functional groove		6		Functional groove		6		Internal plane surface and/or slot		6		Spur gear teeth	
	7			7				Functional cone		7		Functional cone		7		Internal spline (polygon)		7		Bevel gear teeth	
	8			8				Operating thread		8		Operating thread		8		Internal and external polygon, groove and/or slot		8		Other gear teeth	
	9			9				All others		9		All others		9		All others		9		All others	

Figura 3.4 - Código para cada dígito do modelo Opitz [Askin, 1993].

Considerando a parte descrita na Figura 3.5, será obtido o código 15100 [Lorini, 1993]. A seguir é demonstrado o passo para a geração da codificação:

- tamanho do diâmetro: Dígito um, L/D = 1.5 (código 1);
- diâmetro externo: Dígito dois (código 5);
- diâmetro interno: Dígito três (código 1);
- superfície: Dígito quatro, nenhuma (código 0); e
- outras regras: Dígito cinco, nenhuma (código 0).

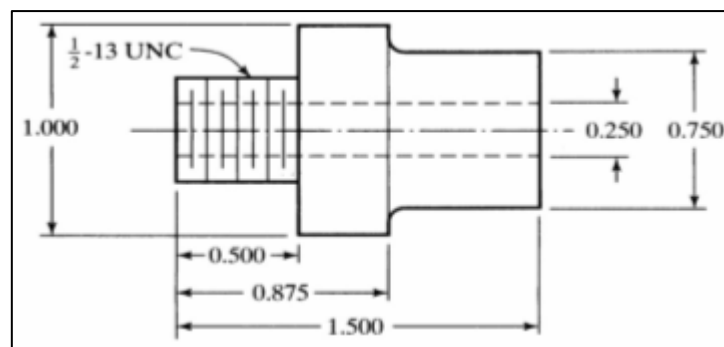


Figura 3.5 - Projeto da parte para o modelo Opitz [Lorini, 1993].

3.4.2.e.2 MULTICLASS

O MULTICLASS foi desenvolvido com o propósito de automatizar e padronizar projeto, produção e gerenciamento. Utiliza até 30 dígitos de forma hierarquizada ou estruturas de codificação de árvores de decisão na qual os dígitos atuais dependem dos valores informados anteriormente. O modelo MULTICLASS é implementado em programas de computador que fazem uma série de perguntas e com base nas respostas geram o código [Groover, 2001].

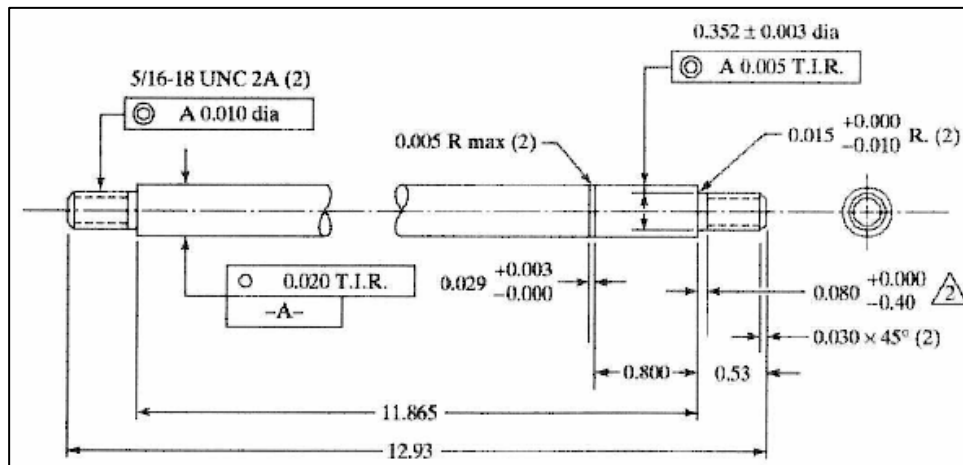


Figura 3.6 - Projeto da parte para o modelo MULTICLASS [Groover, 2001].

A Figura 3.6 apresenta o projeto de uma parte, destacando suas características técnicas para a geração da codificação no modelo MULTICLASS.

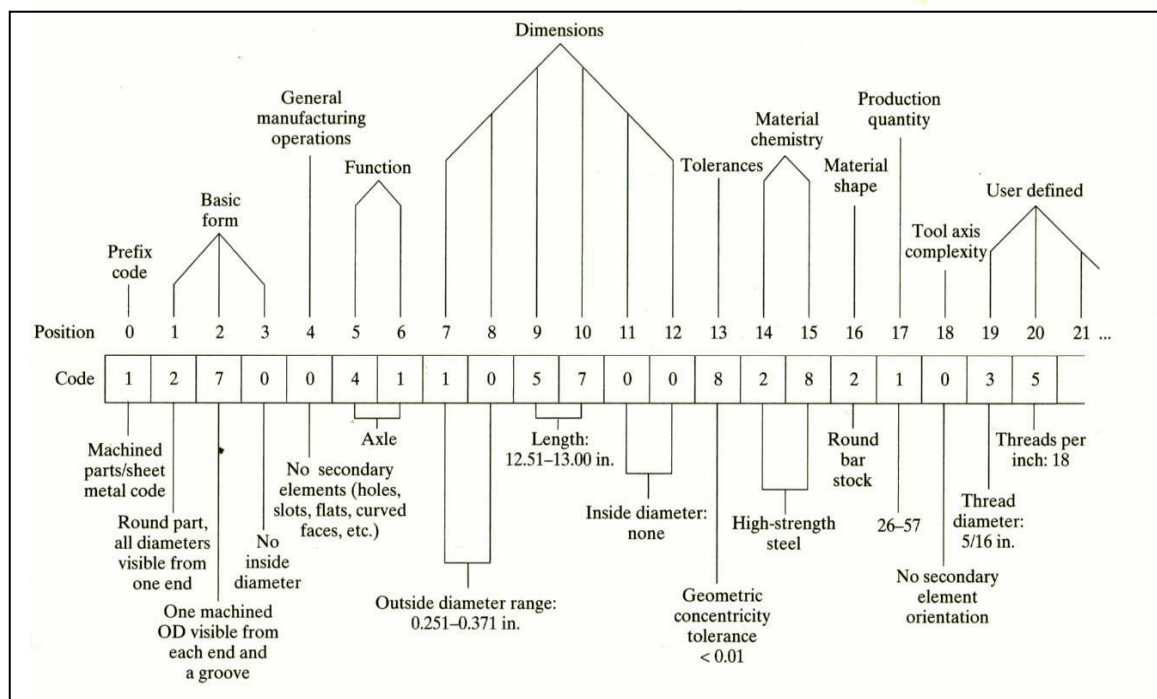


Figura 3.7 - Esquema de codificação MULTICLASS [Groover, 2001].

A codificação para o modelo MULTICLASS é ilustrada na Figura 3.7. Ao aplicar esta codificação para o projeto da Figura 3.6, resultará na seguinte codificação: “0270041105000800200”.

3.4.2.e.3 MICLASS

O sistema MICLASS é um padrão para sistemas de classificação no ramo de atividade metal-mecânico, sendo considerado um dos mais populares sistemas de codificação comerciais dos Estados Unidos [Bedworth, 1991].

O sistema consiste de duas principais sessões, sendo que a primeira é composta por doze dígitos e é utilizada para as características de engenharia e manufatura. A segunda sessão da codificação é opcional, podendo conter cerca de dezoito caracteres.

3.4.2.e.4 KK-3

O KK-3 é um sistema de codificação de propósitos gerais para classificação de peças usinadas, destacando-se processos de corte e retífica. Por volta dos anos 70, foi apresentado o sistema KK-1 contendo 10 dígitos (a versão posterior KK-2 continha os mesmos 10 dígitos). Em 1976 é apresentado o modelo KK-3 composto de 21 dígitos, onde sua codificação era bem mais completa, mantendo a compatibilidade com as suas codificações anteriores [Lorini, 1993]. A Figura 3.8 ilustra o esquema de classificação do KK-3.

Digit	Items	(Rotational component)
1.	Parts name	General classification
2.		Detail classification
3.	Materials	General classification
4.		Detail classification
5.	Major dimensions	Length
6.		Diameter
7.	Primary shapes and ratio of major dimensions	
8.	External surface	External surface and outer primary shape
9.		Concentric screw threaded parts
10.		Functional cut-off parts
11.		Extraordinary shaped parts
12.		Forming
13.		Cylindrical surface
14.		Internal primary shape
15.	Internal surface	Internal curved surface
16.		Internal flat surface and cylindrical surface
17.	End surface	
18.	Nonconcentric holes	Regularly located holes
19.		Special holes
20.	Noncutting process	
21.	Accuracy	

Figura 3.8 - Esquema de codificação do modelo KK-3 [Lorini, 1993].

3.4.3 Análise por Fluxo de Produção

A Análise por Fluxo de Produção (AFP) é um método para identificar FP associadas com um grupo de máquinas utilizando a informação contida nas rotas de produção [Groover, 2001]. Quando a família de partes está identificada, existe a vantagem de produzir partes usando TG em Células de Manufatura (CM). Quando as máquinas são agrupadas, o termo Manufatura Celular¹ é usado para descrever este trabalho de organização [Jha, 1991].

Diversos métodos de agrupamento foram propostos, sendo destacados os seguintes, (a) análise de agrupamentos através de formulação matricial [Burbidge, 1971; King, 1982; Kusiak, 1987], (b) particionamento de grafos [Rajagopalan, 1975; Vannelli, 1986] e (c) programação matemática [Kusiak, 1987b; Choobineh, 1988].

3.4.3.a Formulação Matricial

Na formulação matricial é construída uma matriz de incidência parte versus máquina $M[a_{ij}]$. Esta matriz consiste em entradas binárias (0, 1), onde o valor “1” indica que a máquina “i” é usada no processo pela parte “j”. Quando a matriz inicial é construída, os agrupamentos de partes e máquinas não estão visíveis. Os algoritmos de agrupamento permitem transformar esta matriz inicial de forma a identificar os agrupamentos. Segue abaixo, exemplo de uma matriz inicial e sua transformação em agrupamentos [Jha, 1991].

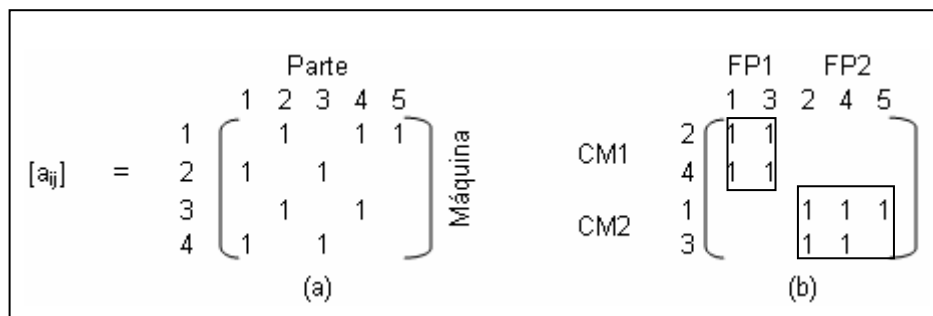


Figura 3.9 - (a) Matriz inicial e (b) Matriz após agrupamento.

O agrupamento através da matriz de incidência pode resultar em duas categorias de agrupamento, (a) mutuamente separados e (b) parcialmente separados [Kusiak, 1990].

¹ Manufatura Celular é a aplicação da TG na qual organiza máquinas ou processos em células. Os objetivos da Manufatura Celular são similares ao da TG [Jha, 1991].

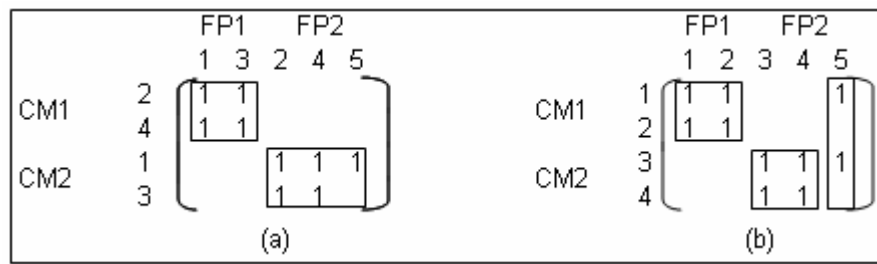


Figura 3.10 - (a) Matriz mutuamente separada e (b) Parcialmente separados.

Na Figura 3.10a ocorre o agrupamento em duas CMs obtendo duas FPs distintas. Contudo, na Figura 3.10b, não pode ocorrer o agrupamento mutuamente separado por causa da parte 5. Para este caso de compartilhamento da parte 5 entre as duas CM, chamamos de agrupamento parcialmente separados [Jha, 1991].

3.4.3.a.1 Medida de Desempenho

Muitas medidas foram propostas para verificar o agrupamento após a matriz ter sido submetida a uma técnica. Existem duas medidas mais populares utilizadas para medir o agrupamento através da geração da diagonalização dos blocos da matriz, sendo elas: (a) eficiência e a (b) eficácia [Gonçalves, 2004].

O agrupamento por eficiência foi proposto por [Chandrasekharan, 1989] e incorpora tanto a utilização da máquina como o movimento entre as células, e é definido pelo somatório dos pesos de duas funções, tal que:

$$\text{Eficiência} = E = qE_1 + (1 - q)E_2 \quad (3.1)$$

Onde,

E_1 = É a média de números 1's do bloco diagonal pelo número total de elementos no bloco diagonal da matriz final;

E_2 = É a média de números 0's do bloco não-diagonal pelo número total de elementos no bloco não diagonal da matriz final;

q = Fator de peso.

A eficiência segundo Gonçalves [Gonçalves, 2004], possui capacidade muito baixa para distinguir uma boa qualidade de agrupamento da ruim. Uma solução ruim com muitos 1's no bloco não-diagonal resulta uma eficiência em torno de 75% conforme matriz da Figura

3.11a. Quando aumenta o tamanho da matriz, o efeito de 1's no bloco não-diagonal torna-se menor, e em alguns casos o efeito de movimento entre as células não reflete a eficiência.

		Máquina																				
Produto	1	2	3	4	5	6	7	8	9	10	11	12										
1	1			1																		
2		1							1													
3			1				1		1													
4	1			1																	1	
5				1				1		1												
6	1			1																	1	
7				1					1													
8				1						1												
9				1					1													
10					1								1								1	
11													1									1
12																					1	
13	1																					1
14						1		1														1
15						1		1														1

(a)

		Máquina																				
Produto	4	8	3	6	7	10	12	1	4	11	2	9										
3	1	1	1																			
5	1	1	1																			
7		1	1																			
9	1	1	1																			
10																						
14						1				1		1										
15						1				1		1										
1																					1	1
4																					1	1
6																					1	1
12																					1	1
13																					1	1
2																						1
8																						1
11																						1

Figura 3.11 - (a) Matriz inicial e (b) Matriz final.

Kumar [Kumar, 1990] propôs uma outra medida de desempenho chamada de eficácia. A eficácia não possui alteração caso o tamanho da matriz mude e é definida como:

$$\text{Eficácia} = \mu = \frac{N_1 - N_1^{out}}{N_1 + N_0^{in}} \quad (3.2)$$

Onde,

N_1 = Número total de 1's na matriz;

N_1^{out} = Número total de 1's fora do bloco diagonal;

N_0^{in} = Número total de 0's dentro do bloco diagonal;

Com base nas Figuras 3.11a e 3.11b é possível demonstrar o desempenho da eficácia do agrupamento.

$$\mu_a = \frac{39 - 0}{39 + 141} = 21,67\% \quad (3.3)$$

$$\mu_b = \frac{39 - 0}{39 + 6} = 86,67\% \quad (3.4)$$

A partir dos resultados obtidos acima, constata-se a grande diferença de desempenho do agrupamento medido através da eficácia, considerando uma matriz que não ocorreu agrupamento (Figura 3.11a) e outra que ocorreu (Figura 3.11b) [Gonçalves, 2004].

3.4.3.a.2 Trabalhos relacionados

Um dos trabalhos de maior importância, principalmente pelo seu pioneirismo, foi apresentado por Burbidge [Burbidge, 1971], em que estabelece a representação da matriz parte versus máquina.

McAuley *apud* [Lorini, 1993] aplicou um coeficiente de similaridade para utilização como parâmetro de comparação entre as máquinas para geração de CM conhecido como *Single Linkage Cluster Analysis* (SLCA).

Outra abordagem é apresentada por McCormick *apud* [Jha, 1991] conhecida como *Bond Energy Algorithm* (BEA), em que a otimização da soma de todos os produtos de cada elemento da matriz parte versus máquina, com seus respectivos elementos mais próximos no arranjo, através da troca nas posições das linhas e colunas.

King *apud* [Jha, 1991] desenvolveu um método de agrupamento denominado de *Rank Order Clustering* (ROC), onde ocorria a ordenação das linhas e colunas da matriz através de uma palavra binária e seu valor decimal correspondente.

Askin *apud* [Jha, 1991] desenvolveu um algoritmo de agrupamento baseado em custo. O algoritmo considera os seguintes custos: (a) custo fixo e variável de máquina, (b) custo de *setup*, (c) custo do ciclo de produção, (d) custo do trabalho em processo e (e) custo de manuseio de material.

Kusiak [Kusiak, 1990] desenvolveu dois algoritmos para a identificação de agrupamentos denominado de CI (*Cluster Identification*) e ECI (*Extended Cluster Identification*), que permite criar agrupamentos exclusivos através da matriz binária partes versus máquina levando em conta um custo associado à parte.

Gómez [Gómez, 1996] desenvolveu um algoritmo que utiliza uma matriz partes versus ferramentas. As abordagens anteriores buscam o bloco diagonal da matriz, porém, nesta abordagem o bloco diagonal deixa de existir, passando a apresentar compartilhamento de recursos entre as partes e ferramentas, buscando respeitar a capacidade do magazine.

Kandiller [Kandiller, 1998] utilizou a representação por hipergrafos que, através de cortes entre todos os pares de vértices, produziu uma árvore de partição e, conseqüentemente, a formação da célula de manufatura e a Família de Parte.

Chen [Chen, 1999] utilizou uma abordagem por decomposição (decomposição de *Bender*) da matriz, e tem como objetivo resolver problemas de grande escala. Basicamente é decomposto o problema original (matriz) em vários subproblemas e então é resolvido cada subproblema separadamente.

Arzi [Arzi, 2001] desenvolveu dois modelos, onde o primeiro modelo baseia-se em programação inteira e aplica-se em problemas de pequenas e médias escalas e, um segundo modelo, na qual utiliza algoritmos genéticos para a resolução de problemas de grande escala.

Xambre [Xambre, 2003] propôs um modelo de programação matemática que visa a minimização do fluxo entre as células de manufatura. Nesta abordagem foi utilizada a meta-heurística *Simulated Annealing* para guiar na otimização proposta e por permitir o controle do tamanho e o número de células geradas.

Gonçalves [Gonçalves, 2004] apresentou uma abordagem que obtém células de manufatura e Famílias de Partes através da combinação de uma heurística de busca local com algoritmo genético.

Rogers [Rogers, 2005] abordou o problema do agrupamento de partes através de um modelo baseado em algoritmos genéticos. Esta abordagem através de algoritmos genéticos visa a resolução de problemas de grande escala. A função objetivo deste modelo busca minimizar o agrupamento simultâneo das partes e máquinas.

Nsakanda [Nsakanda, 2006] desenvolveu uma formulação de célula de manufatura baseada em diversas características da manufatura, tais como: seqüência operacional, demanda da parte, capacidade da máquina, múltiplos planos de processo e múltiplas rotas. Para resolver o problema de formação de célula (Famílias de Partes) foi utilizado um *Framework* de algoritmos genéticos em conjunto com outras abordagens que tratam problemas de grande escala como decomposição por custo.

4 Sistema de Manufatura Flexível

Inicialmente, os Sistemas de Manufatura Flexível (SMF) foram conceituados a partir do desenvolvimento de um Controle Numérico (CN) [Askin, 1993]. Esta concepção foi atribuída ao engenheiro mecânico *David Williamson* em 1960 [Lorini, 1993; Groover, 2001]. Este sistema foi chamado na época de Sistema 24, porque acreditava-se que este sistema operava 24 horas por dia, sendo que 16 horas sem acompanhamento de recursos humanos. Um dos primeiros SMF instalados nos Estados Unidos foi um sistema de máquinas desenvolvido pela *Ingersoll-Rand Company in Roanoke*, na Virgínia por volta de 1960. Os SMF começaram a ser instalados em outros países, como por exemplo: na Alemanha em 1969 e na antiga União Soviética em 1972, obtendo um crescimento industrial mundial [Groover, 2001]. A seguir são apresentadas as definições, o enfoque à flexibilidade que é um dos fatores motivantes deste trabalho, a qualificação dos SMF, seus componentes, benefícios e o modelo conceitual adotado para o SMF que por sua vez será a base para a construção do modelo proposto.

4.1 Definição

Os Sistemas de Manufatura Flexível² (SMF) podem ser definidos como o mais alto nível da TG numa célula de produção, composta por estações de trabalho (CNC), conectadas por um sistema de manuseio de material e um sistema de armazenamento, controlados por um sistema computacional [Groover, 2001].

Segundo Lorini [Lorini, 1993], os SMF podem ser definidos como:

Uma combinação de equipamentos, sistemas de controle e de comunicação integrados na manufatura, para um desempenho de alta produtividade, com capacidade de respostas de modo rápido e econômico a mudanças no ambiente operacional.

² O termo Flexível está na capacidade de processar uma variedade de diferentes estilos de partes simultaneamente em várias estações de trabalho, sendo que este *mix* de estilos de partes e suas quantidades podem ser ajustadas em função de demandas do mercado [Askin, 1993]

Em Askin [Askin, 1993] é definido o FMS como um conjunto de CNCs e estações de trabalho conectadas por um sistema de manuseio de material, todos controlados por um sistema computacional central. Além desta estrutura, incluem alguns elementos do tipo, (a) máquinas de reprogramação automatizadas, (b) troca de ferramenta automatizada, (c) sistema de manuseio de material, tanto para a transferência das partes entre as máquinas como o carregamento/descarregamento de partes nas máquinas e (d) um controle coordenado. A figura abaixo apresenta um exemplo de FMS.

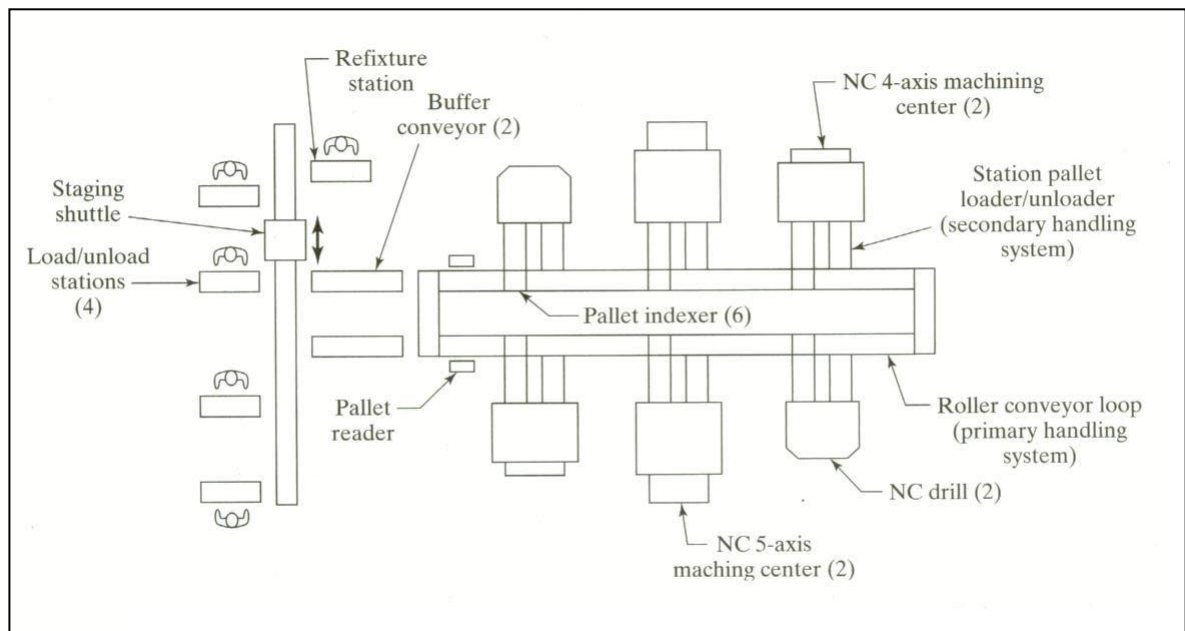


Figura 4.1 - Sistema de Manufatura Flexível [Groover, 2001].

4.2 Flexibilidade

Três são as capacidades que o SMF deve possuir para ser flexível, a habilidade de identificar e distinguir números diferentes de estilos de partes ou produtos a serem processados no sistema, trocas rápidas das instruções de operação e troca rápida de ferramentas [Groover, 2001]. Em Browne [Browne, 1984] são apresentados os tipos de flexibilidade num FMS, conforme tabela abaixo:

Tabela 4.1 – Tipos de flexibilidade num SMF

Tipo de Flexibilidade	Definição
Flexibilidade da Máquina	Capacidade de adaptar a máquina para uma escala maior de operações e partes diferentes de produção.
Flexibilidade de Produção	É o universo de partes diferentes que podem ser produzidas.
Flexibilidade <i>mix</i>	É a habilidade de mudar o <i>mix</i> de produto enquanto se mantém a quantidade de produção total.
Flexibilidade de Produto	Facilidade de adaptação a qualquer mudança no projeto de produtos já existentes bem como novos produtos.
Flexibilidade de Expansão	Facilidade de expandir o sistema com propósito de aumentar a quantidade total de produção.
Flexibilidade de Volume	Habilidade de produzir partes economicamente viáveis, tanto em baixa como alta produção.
Flexibilidade de Roteamento	Capacidade de produzir partes através de células de produção alternativas, em virtude de uma quebra na linha de produção.

Fonte: Browne [Browne, 1984].

4.2.1 Qualificação

Para qualificar a Flexibilidade num SMF, devem ser satisfeitos alguns critérios quanto à automação. A seguir são apresentados quatro testes de Flexibilidade num Sistema de Manufatura Automatizado [Groover, 2001].

- teste da diversidade de partes: o sistema deve ser capaz de processar diferentes estilos de partes sem que este seja em lote;
- teste da reprogramação: o sistema deve permitir mudanças de programação como, por exemplo, um *mix* diferente de partes;
- teste da restauração de erro: o sistema deve ser capaz de produzir continuamente sem que existam paradas de produção ou problemas de funcionamento do mesmo; e
- teste da nova parte: o sistema deve ser capaz de introduzir um novo projeto num *mix* de produtos existentes.

4.2.2 Número de Máquinas

A Flexibilidade num SMF pode ser distinguida também pelo número de máquinas no sistema. Podem ser separados em três categorias [Lorini, 1993; Groover, 2001] conforme ilustrado na Figura 4.1.

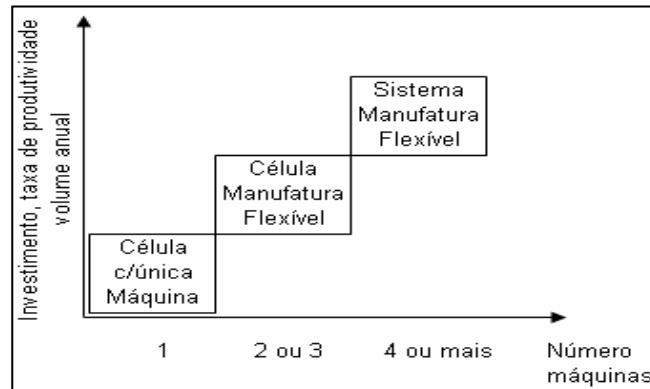


Figura 4.1 - Relação das categorias com a Flexibilidade [Groover, 2001].

- célula com uma única máquina: consiste de uma CNC combinada com um sistema de armazenamento das partes. Esta categoria é capaz de satisfazer três dos quatro testes de qualidade, sendo eles: processamento de diferentes estilos partes, reprogramação e capacidade de introduzir uma nova parte;
- célula de manufatura flexível: consiste de duas ou três estações de trabalho com um sistema de manuseio de parte. Este sistema é conectado com uma estação de carregamento/descarregamento; e
- sistema de manufatura flexível: consiste de pelo menos quatro estações de trabalho interligadas por um sistema de manuseio de parte e são controladas por um sistema computacional.

4.3 Componentes

A composição de um SMF é bastante complexa, pois é composta por máquinas e ferramentas, robôs, sistemas de transporte e um sistema computacional que controla o processo [Lorini, 1993]. A seguir são apresentados os principais componentes deste sistema.

4.3.1 Máquinas

Neste componente, encontram-se diversas máquinas com propósitos gerais e/ou específicos. Por exemplo, em operações de usinagem são utilizados CNC e em outras operações que trabalham com partes prismáticas, a utilização de centros de usinagem horizontais se torna o mais adequado [Askin, 1993].

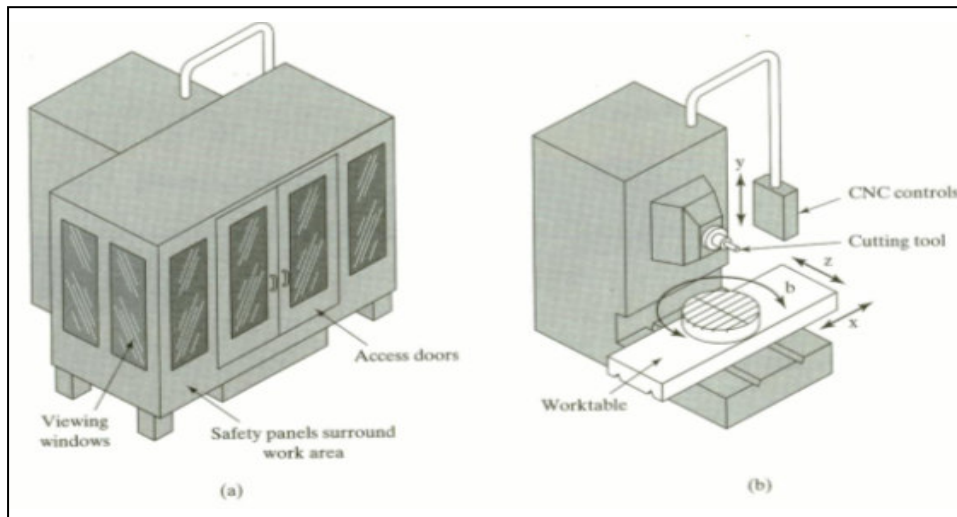


Figura 4.2 - (a) CNC horizontal e (b) CNC sem a proteção (4 eixos) [Groover, 2001].

4.3.2 Estações de carregamento e descarregamento

As estações são de extrema importância no SMF, pois são elas que carregam as partes no sistema e após o término da operação, descarregam em outro processo ou num centro de armazenagem [Lorini, 1993].

4.3.3 Equipamentos para transporte

Para que as partes sejam transportadas entre as estações de carregamento/d Descarregamento, diversos tipos de equipamentos foram desenvolvidos, podendo ser de (a) transporte contínuo, destacando as esteiras, (b) veículos especiais como vagões sobre trilhos e veículos controlados remotamente, e (c) robôs que são capazes de executar tarefas realizadas por recursos humanos [Askin, 1993].

4.3.4 Paletes

Os paletes são componentes de armazenamento das partes processadas. Existem dois tipos, (a) os paletes para lotes pequenos utilizados diretamente em robôs e veículos controlados remotamente, reduzindo assim o deslocamento do próprio palete, e (b) os paletes que acomodam as partes no momento do seu processamento [Lorini, 1993].

4.3.5 Elementos de fixação

Conforme mencionado anteriormente, para acomodar a parte no momento do seu processamento, é necessário fixar a parte no palete. Existem diversos elementos de fixação, sendo estes específicos por parte ou padronizados para atender qualquer parte [Lorini, 1993].

4.3.6 Ferramentas

Num SMF a troca de ferramentas pode ser parte do processo, devido a seu desgaste natural ou até mesmo quebra. As ferramentas podem ser montadas diretamente na máquina e sofrer manutenção automática através de robôs ou serem trocadas manualmente em virtude de uma reprogramação no *mix* de estilos de partes [Lorini, 1993].

4.3.7 Recursos humanos

Mesmo em SMF onde as células são totalmente automatizadas, há utilização de recursos humanos para a execução de uma tarefa específica, tais como: carregamento e descarregamento das partes no sistema, realização do monitoramento e gerenciamento do sistema, manutenção e reparos de equipamentos e troca do conjunto de ferramentas [Askin, 1993; Groover 2001].

4.3.8 Sistema computacional

A gestão da informação da manufatura se dá através de um sistema computacional. É nele onde são passadas todas as instruções para as máquinas, sistemas de manuseio de material, controle e gerenciamento do processo. No sistema computacional é possível controlar as estações de trabalho, realizar a distribuição das instruções de controle para as estações de trabalho, controlar a produção, controlar os movimentos das partes processadas entre as

estações de trabalho, monitorar a situação de cada parte e controlar a ferramenta [Askin, 1993; Lorini, 1993; Groover, 2001].

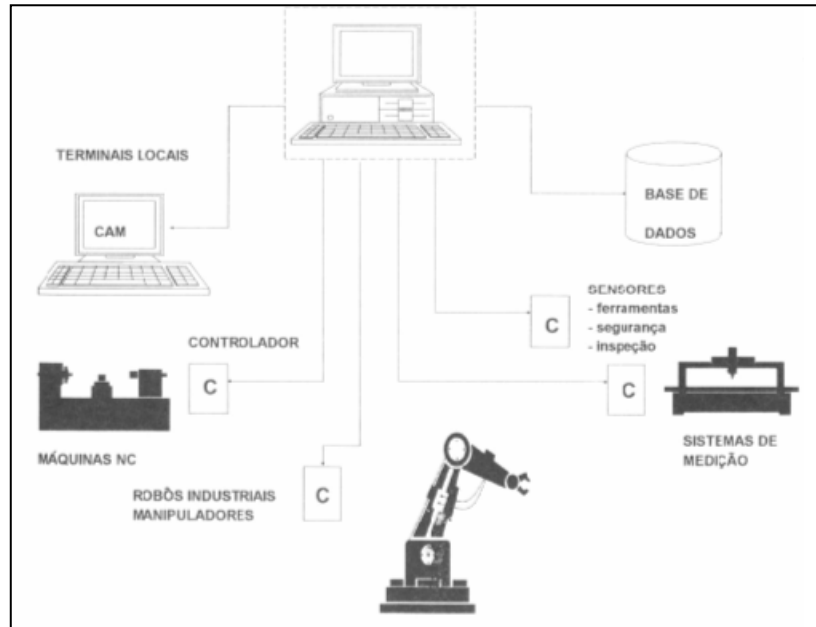


Figura 4.3 - Computador numa célula de manufatura [Lorini, 1993].

4.4 Benefícios

Em Groover [Groover, 2001] são apresentados benefícios na utilização de SMF. A seguir são citados alguns benefícios:

- incremento na utilização da máquina: nota-se que existe uma média de utilização maior que as máquinas convencionais em virtude de alguns aspectos do tipo: operação da máquina 24 horas por dia, troca automática de ferramentas, troca de paletes automatizadas nas estações de trabalho, programação dinâmica, entre outros;
- número inferior de máquinas necessárias: devido a grande utilização das máquinas mencionado no benefício acima, é necessário um número menor de máquinas;
- leiaute reduzido: em virtude da integração entre todos os componentes, o espaço ocupado no chão de fábrica é menor do que os modelos convencionais;
- resposta rápida à mudanças: fornece agilidade e velocidade na introdução de uma nova parte para produzir, paradas de produção, mudança no *mix* de estilos de partes, entre outros;

- redução do inventário: por causa das diferentes partes serem produzidas juntamente, o trabalho em processo é menor do que o modo de produção em lote. O inventário possui de 60% a 80% de redução;
- baixo tempo de produção da parte: em virtude da redução do trabalho em processo, o tempo utilizado para a produção das partes é menor, resultando em entregas rápidas para o cliente;
- alta produtividade: cerca de 30% a 50% do aumento da produtividade são estimados; e
- oportunidade para não acompanhar a produção: o alto nível de automação do SMF permite por longos períodos de produção a não-utilização de recursos humanos.

4.5 Modelo conceitual

Esta proposta utiliza um modelo de SMF apresentado por Stecke *apud* [Gómez, 1996] e que possui quatro níveis de decisão dispostos de forma hierárquica, sendo eles:

- projeto (planejamento estratégico): consiste na seleção das partes a serem produzidas, a escolha das máquinas e ferramentas utilizadas pela parte, a disposição e organização das máquinas (planta), os sistemas de transporte, entre outros;
- planejamento do processo (planejamento tático): consiste na utilização eficiente de recursos, tais como máquinas, ferramentas e sistemas de carregamento (organização da produção);
- escalonamento da produção: consiste na definição de uma seqüência em função do tempo de forma a otimizar suas operações de montagem ou processo; e
- controle e monitoramento: consiste no controle e monitoramento da produção em tempo real de forma a gerenciar a produção.

Os dois primeiros níveis do modelo estão relacionados a fase pré-operacional do SMF, considerando aspectos do planejamento, a organização das partes e ferramentas antes que o sistema comece o processo de produção.

Já os dois últimos níveis são denominados de fase operacional, que por sua vez está relacionado ao escalonamento e rota das partes quando o sistema está em produção. A Figura 4.4 apresenta uma aplicação do modelo hierárquico desenvolvida por Sodhi *apud* [Gómez, 1996].

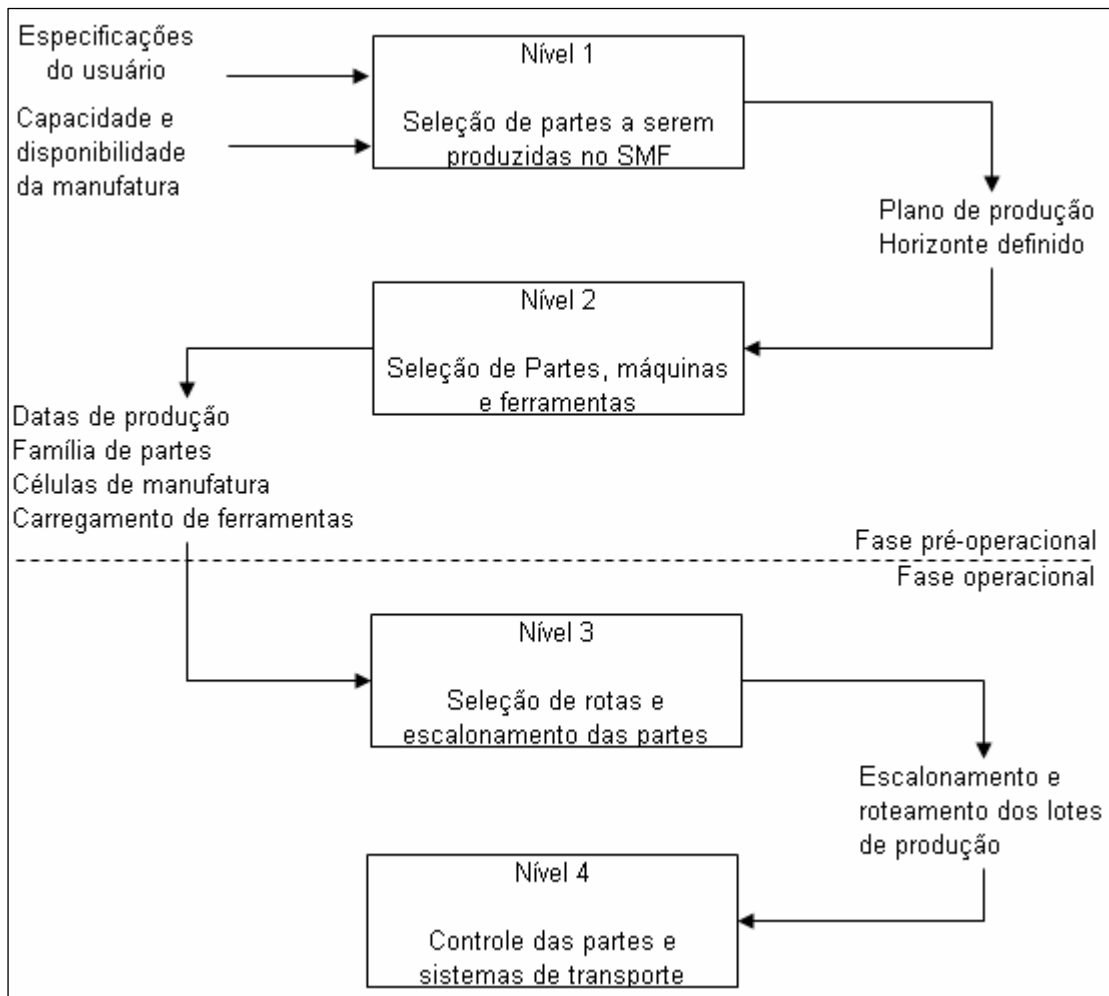


Figura 4.4 - Aplicação do modelo hierárquico.

Stecke *apud* [Gómez, 1996] propõe para a fase pré-operacional, a subdivisão em cinco problemas distintos, sendo eles: (a) seleção de partes, o (b) agrupamento de máquinas para realizar operações similares, a (c) proporção de produção em relação aos tipos de partes a serem produzidas, a (d) alocação de recursos (ferramentas) de modo a limitar o número deste recurso, e o (e) carregamento das partes nas máquinas através das rotas que cada parte deve realizar, respeitando as restrições de capacidade. Já na fase operacional, são considerados os problemas de escalonamento, rotas e controle das partes no processo, sendo que o problema do escalonamento é uma tarefa de difícil resolução [Gómez, 1996].

5 Problemas Abordados

Neste capítulo são apresentados dois problemas que são abordados na geração do modelo proposto aplicado ao JSSP. O primeiro problema está relacionado com a fase pré-operacional e consiste na Seleção de Partes. Já o segundo problema é abordado na fase operacional, e diz respeito ao Escalonamento das Partes, respeitando a capacidade fabril (máquinas e turnos de produção) e as datas de entrega Sodhi *apud* [Gómez, 1996].

5.1 Seleção de Partes

A partir do estado da arte apresentado no capítulo 3, nota-se o esforço no desenvolvimento e aperfeiçoamento de diversos métodos para a geração da FP, sendo destacados a Inspeção Visual, Classificação por Codificação e Análise por Fluxo de Produção (formulação matricial, programação matemática e particionamento de grafos).

A Seleção de Partes é resolvida através do agrupamento das partes, baseada nas similaridades, tais como: forma geométrica, processos de produção semelhantes, similaridade por um mesmo conjunto de ferramentas, entre outros [Kusiak, 1992; Jha, 1991]. O problema da Seleção de Partes é NP-Difícil [Kusiak, 1987].

Em indústrias, na existência de um grande número de estilos de partes e um grande número de operações, ocorre a baixa produtividade do SMF em virtude da flexibilidade. Através da Seleção de Partes é possível obter um processamento simultâneo por uma FP, diminuindo assim o grande número de estilos de partes a ser tratado pelo sistema [Kusiak, 1992].

Em consequência de diversos trabalhos de geração de FP através da Análise por Fluxo de Produção apresentado no capítulo 3, será utilizada a formulação matricial como técnica para a Seleção de Partes e sua contextualização será realizada no capítulo 6.

5.2 Escalonamento de Partes

O problema do escalonamento tem sido pesquisado desde 1950 e possui aplicação em diversas áreas, tais como: manufatura, hospitais, logística, agricultura, etc. O escalonamento consiste em alocar recursos de forma a ordená-los em tarefas a serem executadas. Estes

recursos são limitados de várias maneiras, dentre elas o tempo disponível. A complexidade para este problema é NP-Difícil [Garey, 1976; Leung, 2004].

O escalonamento das partes pode ser definido através de diversos objetivos, dependendo da realidade e foco da indústria. O objetivo principal é a minimização do tempo total de produção (*makespan*). Contudo, pode-se utilizar outros objetivos, tais como: redução do trabalho em processo (*work in process*), minimizar as trocas de ferramentas, minimizar o tempo de atraso, maximizar a eficiência, entre outros [Jain, 1998].

A divisão para o problema do escalonamento obedece três modelos [Blazewicz, 2002]:

- *flow-shop*: consiste na produção contínua. Caracteriza-se pelas linhas de produção, ou seja, o fluxo do processamento das partes é linear. O *mix* de partes é muito pequeno (produção em massa);
- *open-shop*: é um ambiente aberto. O número de operações é igual ao número de máquinas, contudo, a ordem de processamento não é especificada; e
- *job-shop*: o fluxo de produção é realizado de acordo com a rota de produção definida na parte. Caracteriza-se por um *mix* de partes alto e fluxo de produção variado.

5.2.1 O Problema do Escalonamento no Job-shop

Em Blazewicz, Jain e Zoghby [Blazewicz, 1996; Jain, 1998; Zoghby, 2004], é definido o *Job-shop Scheduling Problem* (JSSP) como um conjunto de “n” *jobs* a serem processados num conjunto de “m” máquinas. Cada *job* é composto de “i” operações que devem ser processadas através de uma rota; para cada operação é definida uma máquina com tempo de processamento padrão. O objetivo é minimizar o tempo total de produção (*makespan*).

Já Gómez [Gómez, 1996] apresenta a dificuldade em programar a produção de num *Job-shop* com alta produtividade, em virtude do *mix* de estilos de partes, mudanças no mercado e a flexibilidade do SMF. Segundo Rantakyro [Rantakyro, 2000] a tarefa em gerenciar a produtividade num *Job-shop* é definida como:

A produção é muito difícil de ser planejada porque existem muitos clientes e uma grande quantidade de diferentes produtos a serem manufaturados. As empresas concorrem em todos os tipos de produtos e eles são produzidos em lotes muitos pequenos, às vezes o tamanho do lote é unitário.

A maioria dos problemas de escalonamento estudados aplica-se ao *Job-shop* e é caracterizado por permitir diferentes *jobs* entre as máquinas e diferentes números de operações por *job*, que são processadas apenas uma vez em cada máquina. Também podemos dizer que é uma forma de produção onde as diversas partes a serem produzidas atravessam o sistema de manufatura através de rotas diferentes [Slack, 1997].

Existem diversos subproblemas propostos para o JSSP, onde podem ser destacados os *Job-shops* preemptivos (o processo pode ser interrompido e reiniciar a partir do ponto em que parou), não-preemptivos (o processo não é interrompido), com restrições às datas de entrega, com operações de gargalos, entre outros. Neste trabalho é utilizado um ambiente de JSSP não-preemptivo com restrições às datas de entrega considerando a Manufatura Flexível. A figura a seguir ilustra o gráfico *Gantt* para o tipo de JSSP proposto.

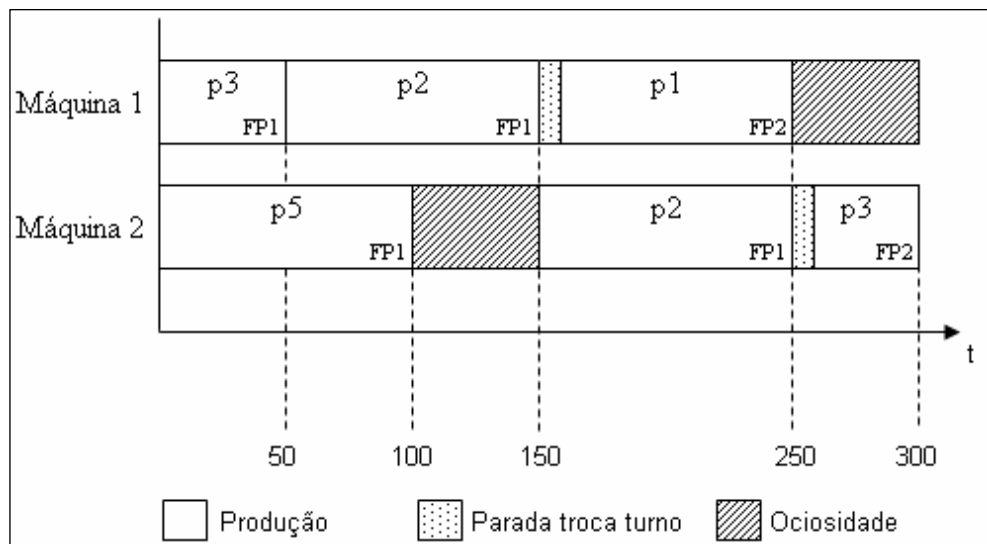


Figura 5.1 - Escalonamento do *Job-shop* com paradas de produção e ociosidade.

5.2.1.a Complexidade

Segundo Garey [Garey, 1976] o JSSP num SMF é uma tarefa de difícil complexidade devido a este problema ser conhecido na teoria da complexidade como NP-Difícil. O problema NP-Difícil refere-se às classes de problemas de decisão que contém todos os problemas “h”, tal que para todos os problemas “l” em NP existe uma redução em tempo polinomial para “h”, ou seja, é uma classe que contém problemas de decisão que pelo menos são tão difíceis quanto um problema NP [Garey, 1979]. A seguir, é apresentado um exemplo da relação de crescimento da complexidade do problema em relação ao número de entradas [Kusiak, 1994].

Supondo-se que, para oito *jobs* serem seqüenciados em três máquinas, o número de seqüências possíveis seria de $8!^3 \approx 65,000,000,000,000$. Se um computador realizasse o

processamento de uma seqüência em 0.1 msecs, levaria 206 anos para gerar todas as possibilidades de seqüenciamento.

5.2.1.b Formulação Clássica

Existem diferentes formulações para o JSSP, porém, será adotado o modelo de formulação de Adams *apud* [Blazewicz, 1996], onde:

$V = \{0, 1, \dots, n\}$ representa o conjunto de operações, onde “0” é a primeira operação de todos os *jobs* e “n” será a última operação para todos os *jobs*. O conjunto de “m” máquinas é representado por “M” e “A” é a representação para o conjunto de pares ordenados das restrições de operações pela precedência das relações de cada *job*. Para cada máquina “k”, o conjunto de “ E_k ” descreve todos os pares de operações fornecidos pela máquina “k”. Para cada operação “i” é processado num tempo “ p_i ” (fixo) e o processo inicial de “i” é “ t_i ”, uma variável que tem sido determinada durante a otimização. A partir destas definições, o modelo está contextualizado da seguinte forma:

$$\min t_n \quad (5.1)$$

$$t_j - t_i \geq p_i \quad \forall (i, j) \in A, \quad (5.2)$$

$$t_j - t_i \geq p_i \text{ ou } t_i - t_j \geq p_j \quad \forall \{i, j\} \in E_k, \quad (5.3)$$

$$\forall k \in M, \quad (5.4)$$

$$t_i \geq 0 \quad \forall i \in V.$$

A função objetivo (5.1) para o JSSP busca minimizar o tempo total de produção. A restrição (5.2) assegura que a seqüência de processamento das operações para cada *job* corresponde a uma ordem pré-determinada. Já a restrição (5.3) é a demanda que existe, ou seja, somente um *job* em cada máquina num determinado tempo, e a restrição (5.4) assegura o término de todos os *jobs*.

Para o modelo apresentado, é necessário fazer algumas restrições [Jain, 1998; Blazewicz, 1996]:

- não existe regra de precedência entre as operações de diferentes partes;
- as operações que iniciarem o processo não podem ser interrompidas;
- uma máquina pode somente processar um *job* por vez; e
- um *job* pode somente ser processado em uma máquina por vez.

Uma vez que as rotas de máquinas para cada *job* são fixas, o problema a ser resolvido consiste em determinar as rotas dos *jobs* em cada máquina, tal que o tempo de execução transcorrido, desde o início do primeiro *job* até o término do último, seja mínimo, objetivando a solução com menor tempo de produção (*makespan*) [Jain, 1998].

Normalmente o número de restrições é muito grande, o que torna o escalonamento do JSSP um dos problemas de difícil solução devido a sua complexidade combinatória. Testes para problemas maiores são mais difíceis de obter o escalonamento ótimo conhecido, ou seja, mais difícil é obter o arranjo de todas as tarefas nas máquinas que satisfazem as restrições de precedência cujo processamento se faz em um menor tempo [Zoghby, 2004].

5.2.1.c Exemplo de JSSP

O JSSP é normalmente referenciado como um n/m JSSP, onde “ n ” é o número de *jobs* e “ m ” o número de máquinas [Zoghby, 2004]. A figura a seguir, apresenta o escalonamento para um 3/3 JSSP.

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 2px;">Job 1</td><td style="padding: 2px;">1,1,2</td><td style="padding: 2px;">1,2,2</td><td style="padding: 2px;">1,3,1</td></tr> <tr><td style="padding: 2px;">Job 2</td><td style="padding: 2px;">2,1,3</td><td style="padding: 2px;">2,2,2</td><td style="padding: 2px;">2,3,1</td></tr> <tr><td style="padding: 2px;">Job 3</td><td style="padding: 2px;">3,1,2</td><td style="padding: 2px;">3,2,1</td><td style="padding: 2px;"></td></tr> </table> <p style="text-align: center;">(a)</p>	Job 1	1,1,2	1,2,2	1,3,1	Job 2	2,1,3	2,2,2	2,3,1	Job 3	3,1,2	3,2,1		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="padding: 2px;">Operação</th> <th style="padding: 2px;">Tempo (t)</th> </tr> </thead> <tbody> <tr><td style="padding: 2px;">1,1,2</td><td style="padding: 2px;">5</td></tr> <tr><td style="padding: 2px;">1,2,2</td><td style="padding: 2px;">7</td></tr> <tr><td style="padding: 2px;">1,3,1</td><td style="padding: 2px;">7</td></tr> <tr><td style="padding: 2px;">2,1,3</td><td style="padding: 2px;">7</td></tr> <tr><td style="padding: 2px;">2,2,2</td><td style="padding: 2px;">5</td></tr> <tr><td style="padding: 2px;">2,3,1</td><td style="padding: 2px;">5</td></tr> <tr><td style="padding: 2px;">3,1,2</td><td style="padding: 2px;">7</td></tr> <tr><td style="padding: 2px;">3,2,1</td><td style="padding: 2px;">5</td></tr> </tbody> </table> <p style="text-align: center;">(b)</p>	Operação	Tempo (t)	1,1,2	5	1,2,2	7	1,3,1	7	2,1,3	7	2,2,2	5	2,3,1	5	3,1,2	7	3,2,1	5	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Máquina 1</td> <td style="padding: 2px;">ocioso</td> <td style="padding: 2px;">3,2,1</td> <td style="padding: 2px;">ocioso</td> <td style="padding: 2px;">2,3,1</td> <td style="padding: 2px;">ocioso</td> <td style="padding: 2px;">1,3,1</td> </tr> <tr> <td style="padding: 2px;">Máquina 2</td> <td style="padding: 2px;">3,1,2</td> <td style="padding: 2px;">1,1,2</td> <td style="padding: 2px;">2,2,2</td> <td style="padding: 2px;">1,2,2</td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">Máquina 3</td> <td style="padding: 2px;">2,1,3</td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> </tr> </table> <p style="text-align: center;">(c)</p>	Máquina 1	ocioso	3,2,1	ocioso	2,3,1	ocioso	1,3,1	Máquina 2	3,1,2	1,1,2	2,2,2	1,2,2			Máquina 3	2,1,3					
Job 1	1,1,2	1,2,2	1,3,1																																																		
Job 2	2,1,3	2,2,2	2,3,1																																																		
Job 3	3,1,2	3,2,1																																																			
Operação	Tempo (t)																																																				
1,1,2	5																																																				
1,2,2	7																																																				
1,3,1	7																																																				
2,1,3	7																																																				
2,2,2	5																																																				
2,3,1	5																																																				
3,1,2	7																																																				
3,2,1	5																																																				
Máquina 1	ocioso	3,2,1	ocioso	2,3,1	ocioso	1,3,1																																															
Máquina 2	3,1,2	1,1,2	2,2,2	1,2,2																																																	
Máquina 3	2,1,3																																																				

Figura 5.2 - Solução de escalonamento para o 3/3 JSSP.

A Figura 5.2a representa cada *job* com suas operações (a, b, c), sendo que “a” é o *job*, “b” é o número da operação e “c” a máquina em que a operação deverá ser executada. Já a Figura 5.2b exibe os tempos de processamento para cada operação em unidades de tempo “t”. Por último, a Figura 5.2c apresenta uma solução possível de escalonamento para o 3/3 JSSP.

5.2.1.d Trabalhos relacionados

Por volta da década de setenta, uma das principais preocupações relacionadas ao problema do JSSP era justificar a natureza de complexidade do problema [Garey, 1979; Jain, 1998; Leung, 2004]. Contudo, foi na década de oitenta e noventa que foram desenvolvidos diversos

algoritmos inovadores para a solução do JSSP [Blazewics, 1996; Mascis, 2002; Zoghby, 2004]. São destacados os algoritmos de otimização a passos largos Martin *apud* [Jain, 1998], algoritmos aproximativos de busca no espaço denominado meta-heurísticas, como a Busca Tabu [Nowicki, 1996; Glover, 1997; Tsai, 1998; Hurink, 2004], Têmpera Simulada Aarts *apud* [Blazewics, 1996], Algoritmos Genéticos Davis *apud* [Blazewics, 1996], Algoritmos Genéticos de Busca Local Johnson *apud* [Jain, 1998] e o algoritmo GRASP (*Greedy Randomized Adaptive Procedure*) Resende *apud* [Jain, 1998].

Em Jain [Jain, 1998] é realizado um *survey* dos métodos utilizados para a solução do JSSP. A Figura 5.3 mostra a classificação desses métodos.

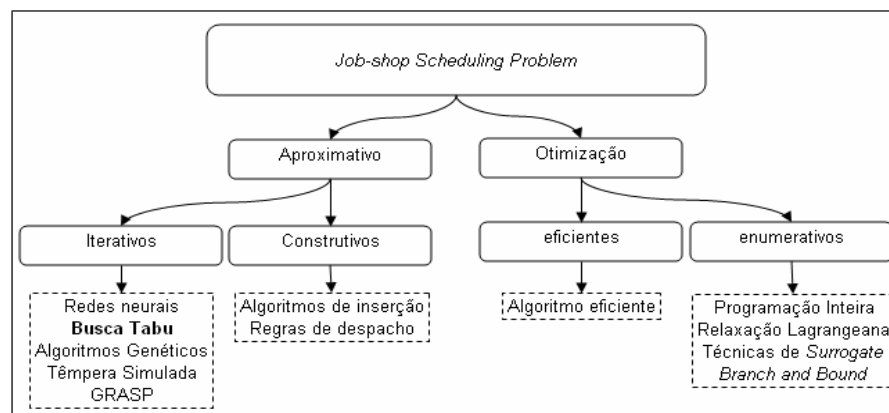


Figura 5.3 - Classificação dos Métodos no JSSP [Jain, 1998].

Inúmeros métodos de otimização foram desenvolvidos para a solução do JSSP, sendo destacados os métodos de otimização e os métodos aproximativos. Para os métodos de otimização são citados a Programação Inteira, a Relaxação Lagrangeana, as Técnicas de *Surrogate* e o *Branch and Bound*. Já nos métodos aproximativos são destacados os algoritmos iterativos Busca Tabu, as Redes Neurais, os Algoritmos Genéticos, a Têmpera Simulada e o GRASP.

O capítulo a seguir, apresenta as técnicas utilizadas para a resolução do problema de Seleção de Partes e do Escalonamento.

6 Técnicas utilizadas

Neste capítulo são apresentadas as duas técnicas utilizadas no modelo aplicado ao JSSP. Sendo apresentado na seção 6.1 o algoritmo *Cluster Identification* (CI) [Kusiak, 1987] utilizado para o problema da Seleção de Partes e, para o problema do Escalonamento, é apresentada na seção 6.2 a meta-heurística Busca Tabu [Glover, 1997].

6.1 Técnica para a Seleção de Partes

Nesta seção é apresentada a técnica para a Seleção de Partes baseada na Análise por Fluxo de Produção utilizando a formulação matricial. Em Kusiak [Kusiak, 1987] é desenvolvido um algoritmo para a Seleção de Partes denominado *Cluster Identification* (CI).

Este algoritmo objetiva o bloco diagonal da matriz, conforme mostrado na Figura 6.1 [Kusiak, 1987; Jain, 1998]. A formulação é descrita no capítulo 3.

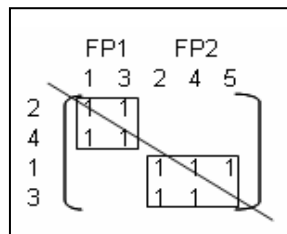


Figura 6.1 - Matriz com bloco na diagonal.

6.1.1 Algoritmo

- Passo 0. Atualizar $k = 1$;
- Passo 1. Selecionar qualquer linha “i” da matriz incidente $A^{(k)}$ e traçar uma linha horizontal “ h_i ” sobre ela;
- Passo 2. Para cada entrada “1” na matriz $A^{(k)}$ cruzada pela linha horizontal “ h_i ”, traçar uma linha vertical “ v_j ”;
- Passo 3. Para cada entrada “1” na matriz $A^{(k)}$ cruzada pela linha vertical “ v_j ”, traçar uma linha horizontal “ h_k ”;
- Passo 4. Repetir os passos 2 e 3 até que não encontre nenhuma entrada “1” na matriz $A^{(k)}$ sem estar cruzada;
- As linhas horizontais e verticais traçadas corresponderão à FP;
- Passo 5. Transformar a matriz incidente $A^{(k)}$ em $A^{(k+1)}$, removendo as linhas e colunas traçadas nos passos 2, 3 e 4;
- Passo 6. Se a matriz $A^{(k+1)} = 0$, pare, pois os elementos são todos zeros; caso contrário faça $k = k + 1$ e volte ao passo 1.

Figura 6.2 - Algoritmo de Identificação de FP [Kusiak, 1987].

6.1.2 Exemplo

Em Jha [Jha, 1991] é descrito um exemplo do Algoritmo CI considerando a matriz de incidência parte versus máquina da Figura 6.3.

$$A = \begin{array}{c} \begin{array}{cccccccc} & & & \text{parte} & & & & \\ & & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \end{array} \\ \begin{array}{cccccccc} 1 & & & & & & & & & & \\ 2 & & & & & & & & & & \\ 3 & & & & & & & & & & \\ 4 & & & & & & & & & & \\ 5 & & & & & & & & & & \\ 6 & & & & & & & & & & \\ 7 & & & & & & & & & & \\ \end{array} \\ \begin{array}{cccccccc} \left(\begin{array}{cccccccc} 1 & 1 & & 1 & & & & & & & \\ 1 & & & & & & 1 & & & & \\ & & & 1 & & & & & 1 & & \\ 1 & & & & & & & 1 & & & \\ & & & 1 & & 1 & & & & & 1 \\ & & & & 1 & & & & & & \\ 1 & 1 & & 1 & & & & & & & 1 \\ \end{array} \right) \\ \end{array} \\ \begin{array}{c} \text{Máquina} \end{array} \end{array}$$

Figura 6.3 - Matriz inicial para o Algoritmo CI.

O algoritmo inicia atribuindo o valor “1” para a variável de iteração “k”. A primeira linha da matriz apresentada na Figura 6.3 é selecionada e traçada uma linha horizontal “ h_1 ”. Os resultados obtidos pelos passos um e dois são apresentados na Figura 6.4.

$$A^{(1)} = \begin{array}{c} \begin{array}{cccccccc} & & & & & & & & & & \\ & & & & & & & & & & \\ \end{array} \\ \begin{array}{cccccccc} 1 & & & & & & & & & & \\ 2 & & & & & & & & & & \\ 3 & & & & & & & & & & \\ 4 & & & & & & & & & & \\ 5 & & & & & & & & & & \\ 6 & & & & & & & & & & \\ 7 & & & & & & & & & & \\ \end{array} \\ \begin{array}{cccccccc} \left(\begin{array}{cccccccc} 1 & 1 & & 1 & & & & & & & \\ 1 & & & & & & 1 & & & & \\ & & & 1 & & & & & 1 & & \\ 1 & & & & & & & 1 & & & \\ & & & 1 & & 1 & & & & & 1 \\ & & & & 1 & & & & & & \\ 1 & 1 & & 1 & & & & & & & 1 \\ \end{array} \right) \\ \end{array} \\ \begin{array}{c} h_1 \\ v_2 \quad v_3 \quad v_5 \end{array} \end{array}$$

Figura 6.4 - Matriz após o passo um e dois do Algoritmo CI.

Os passos um e dois resultaram em três linhas verticais (v_2 , v_3 , v_5) contendo cinco novas entradas com o valor “1” da matriz ilustrada na Figura 6.4. Já no passo três, duas linhas horizontais (h_5 , h_7) são traçadas sobre esta matriz, resultando assim na matriz da Figura 6.5.

$$A^{(1)} = \begin{array}{cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} & \left(\begin{array}{cccccccc} 1 & & & & & & & \\ 1 & & & & & & & \\ & & & & & & & \\ 1 & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{array} \right) & \begin{array}{l} h_1 \\ \\ \\ \\ h_5 \\ \\ h_7 \end{array} \\ & \begin{array}{cccc} v_2 & v_3 & & v_5 \end{array} \end{array}$$

Figura 6.5 - Matriz após o terceiro passo do Algoritmo CI.

No quarto passo, o algoritmo traça uma linha vertical (v_8) com base na matriz da Figura 6.5, ou seja, as entradas (5, 8) e (7, 8).

$$A^{(1)} = \begin{array}{cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} & \left(\begin{array}{cccccccc} 1 & & & & & & & \\ 1 & & & & & & & \\ & & & & & & & \\ 1 & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{array} \right) & \begin{array}{l} h_1 \\ \\ \\ \\ h_5 \\ \\ h_7 \end{array} \\ & \begin{array}{cccc} v_2 & v_3 & & v_5 & v_8 \end{array} \end{array}$$

Figura 6.6 - Matriz após o quarto passo do Algoritmo CI.

Após o quarto passo do algoritmo, foram geradas: $CM_1 = \{1, 5, 7\}$ e $FP_1 = \{2, 3, 5, 8\}$. No entanto, ainda existem entradas na matriz com valor “1” que ainda não foram traçadas. A partir daí, o quinto passo do algoritmo transforma a matriz da Figura 6.6 numa matriz “ $k + 1$ ”, permanecendo nesta matriz somente as entradas com valor “1” que não foram traçadas. A Figura 6.7a apresenta a matriz $A^{(2)}$.

$$A^{(2)} = \begin{array}{cccc} & 1 & 4 & 6 & 7 \\ \begin{array}{l} 2 \\ 3 \\ 4 \\ 6 \end{array} & \left(\begin{array}{cccc} 1 & & & \\ 1 & & & \\ & & & \\ 1 & & & \\ & & & \end{array} \right) & \begin{array}{l} h_2 \\ h_4 \end{array} \\ & \begin{array}{cc} v_1 & v_6 \end{array} \end{array} \quad (a)$$

$$A^{(2)} = \begin{array}{cccc} & 1 & 4 & 6 & 7 \\ \begin{array}{l} 2 \\ 3 \\ 4 \\ 6 \end{array} & \left(\begin{array}{cccc} 1 & & & \\ & & & \\ 1 & & & \\ & & & \\ 1 & & & \end{array} \right) & \begin{array}{l} h_2 \\ h_4 \end{array} \\ & \begin{array}{cc} v_1 & v_6 \end{array} \end{array} \quad (b)$$

$$A^{(3)} = \begin{array}{ccc} & 4 & 7 \\ \begin{array}{l} 3 \\ 6 \end{array} & \left(\begin{array}{cc} 1 & \\ 1 & \end{array} \right) & \begin{array}{l} h_3 \\ h_6 \end{array} \\ & \begin{array}{cc} v_4 & v_7 \end{array} \end{array} \quad (c)$$

Figura 6.7 - (a) Matriz $k + 1$ (b) Matriz final $k + 1$ (c) Matriz $k + 2$.

A partir da matriz $A^{(2)}$ são aplicados os passos de um a quatro do algoritmo CI ilustrado na Figura 6.7b, gerando a $CM_2 = \{2, 4\}$ e $FP_2 = \{1, 6\}$. A Figura 6.7c apresenta a última iteração ($A^{(3)}$) do algoritmo CI, onde resulta na $CM_3 = \{3, 6\}$ e $FP_3 = \{4, 7\}$. A Figura 6.8 apresenta a CM e FP geradas a partir do algoritmo.

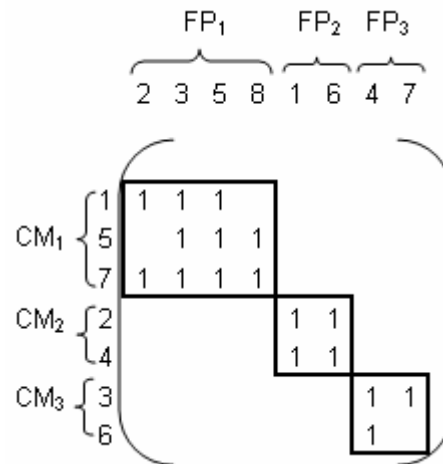


Figura 6.8 - Identificação da CM e FP a partir do algoritmo CI.

6.1.3 Considerações

Segundo Kusiak e Jain [Kusiak, 1987; Jain 1998], as experiências computacionais mostram que o Algoritmo CI é muito eficiente. Considerando uma matriz $m \times n = 60 \times 80$ com valores possíveis de 0's e 1's, o algoritmo obteve a resposta em 0.06 segundos. A complexidade³ do Algoritmo é $O(mn)$, onde “m” é o número de linhas e “n” é o número de colunas.

6.2 Técnica para o Escalonamento das Partes

Nesta seção é apresentado primeiramente o conceito de meta-heurística e, após sua definição, a técnica Busca Tabu.

6.2.1 Meta-heurística

O termo meta-heurística deriva da palavra grega *heurisken* que significa encontrar, enquanto o sufixo “meta” significa além de, em um nível superior [Blum, 2001]. Segundo Glover [Glover, 1997], meta-heurística consiste numa estratégia principal que guia e modifica outras

³ A notação O (*big O*) estabelece um limite superior para a complexidade do algoritmo em função da sua entrada, permitindo assim analisar o desempenho do algoritmo [Garey, 1976].

heurísticas para produzir soluções que são normalmente direcionadas para a solução ótima local.

Em Reeves [Reeves, 1993], uma meta-heurística é uma técnica que procura soluções boas (próximas da ótima) a um custo computacional razoável sem se tornar hábil para garantir a viabilidade ou o grau de otimalidade dessa solução, ou até em muitos casos a meta-heurística pode dizer o quão próxima uma solução viável está da solução ótima. Isso para casos em que a solução ótima já seja conhecida para algumas instâncias do problema em análise.

De forma mais detalhada alguns autores afirmam que a meta-heurística é um processo de geração iterativa que orienta uma heurística subordinada por combinar, de forma inteligente, conceitos diferentes para investigar e explorar o espaço de busca⁴, e utilizar estratégias de memorização para estruturar informações a fim de obter soluções eficientes e próximas do ótimo global. Conforme Viana [Viana, 1998], o termo investigar se refere à capacidade da meta-heurística de saltar, a passos largos, de uma região para outra no espaço de busca. Já o termo exploração reflete a capacidade em explorar de forma mais intensa uma mesma região dentro do espaço de busca. Em resumo, o que existe são dois processos de busca, um externo (investigação) e outro interno (exploração). A investigação é comumente chamada de diversificação, enquanto a exploração é denominada de intensificação.

Entre as principais características apresentadas pelas meta-heurísticas é importante citar:

- conduzem o processo de busca;
- exploram, eficientemente, o espaço de busca a fim de encontrar soluções ótimas;
- variam de procedimentos de busca local simples para processos de memorização complexos;
- são aproximativas e não-determinísticas;
- incorporam mecanismos para evitar com que a busca fique confinada em um determinado local do espaço de busca;
- não são específicas para um único tipo de problema; e
- fazem uso de conhecimento de domínio específico e/ou experiência de busca (memória) para influenciar a busca.

⁴ Considerando uma região com 500 cidades, o espaço de busca compreende a totalidade dessas cidades.

As meta-heurísticas mais encontradas e conseqüentemente as mais aplicadas são: Têmpera Simulada (*Simulated Annealing*), Busca Tabu (*Tabu Search*), Algoritmos Genéticos (*Genetic Algorithms*) e GRASP (*Greedy Randomized Adaptive Procedure*) [Viana, 1998]. Essas meta-heurísticas podem ser aplicadas na resolução de diversos problemas, tais como: caixeiro viajante, carteiro chinês, escalonamento de tarefas, programação inteira, entre outros. Na seção seguinte será discutida a meta-heurística Busca Tabu.

6.2.2 Busca Tabu

A meta-heurística Busca Tabu (BT) teve origem a partir de uma solução para problemas de programação inteira proposta por *Glover* em 1986; posteriormente ele deu uma descrição do método, para uso geral em problemas da área de Pesquisa Operacional (otimização combinatória). Independentemente, *Hansen* também em 1986 propôs uma técnica semelhante, que ele denominou de SAMD (*Steepest Ascent, Mildest Descent*), usada para resolver o problema da satisfiabilidade; entretanto, *Fred Glover* é considerado o criador do algoritmo BT por seus inúmeros trabalhos publicados [Viana, 1998].

Basicamente, a BT, que foi projetada para encontrar boas aproximações para a solução ótima global de qualquer problema de otimização, possui três princípios fundamentais: (a) uso de uma estrutura de dados (fila) para guardar o histórico da evolução do processo de busca, (b) uso de um mecanismo de controle para fazer um balanceamento entre a aceitação, ou não, de uma nova configuração, com base nas informações registradas na lista tabu referentes às restrições e aspirações desejadas e (c) incorporação de procedimentos que alternam as estratégias de diversificação e intensificação. Na figura abaixo, é possível visualizar a lista tabu e as interações entre os componentes de diversificação e intensificação [Glover, 1997; Viana, 1998].

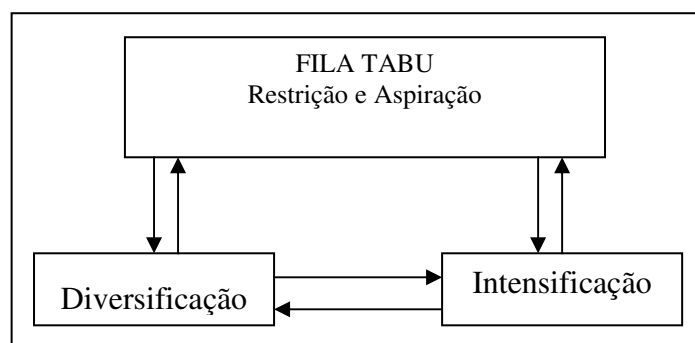


Figura 6.9 - Estratégia de busca da BT [Viana, 1998].

As estruturas de memória apresentadas na figura abaixo, caracterizam quatro principais dimensões: (a) recenticidade: se a solução foi visitada há pouco tempo. (b) frequência: quantas vezes a solução ou atributo foi visitado. (c) qualidade: habilidade de diferenciar o mérito de uma solução durante a busca. Pode ser medida tanto em relação à solução total como em relação a partes da solução. (d) influência: influência de uma determinada solução, ou determinado atributo, em relação às próximas soluções ou atributos [Glover, 1997].

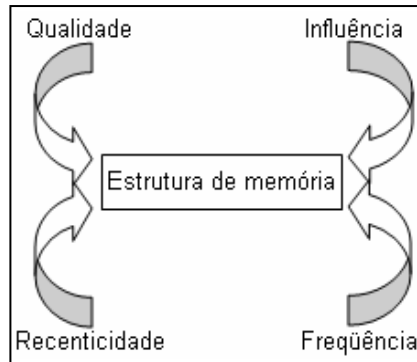


Figura 6.10 - Quatro principais dimensões da BT [Glover, 1997].

Uma descrição geral do algoritmo básico da BT para um problema de minimização (sem diversificação ou intensificação) é apresentada a seguir na Figura 6.11.

6.2.3 Algoritmo

```

s = solução inicial obtida através de uma regra de despacho;
niter = 0;
melhiter = 0;
nbmax = 0;
melhsol = s;
Fmelhor = F(s);
Fmin = valor mínimo estimado da função;
Inicializar a lista Tabu;
Inicializar a função do critério de aspiração  $A(z = F(s))$ ;

ENQUANTO ( $F(s) > Fmin$ ) OU ( $niter - melhiter < nbmax$ ) FAÇA
    niter = niter + 1;
    Gerar um conjunto  $V^*$  de soluções em  $N(s)$ ;
    Escolher a melhor solução  $s^*$  em  $V^*$  que não seja tabu ou  $F(s^*) < A(F(s))$ ;
    Atualizar a função do critério de aspiração  $A(z = F(s))$  e a lista tabu
    retirando o movimento mais antigo;

    SE  $F(s^*) < F(melhsol)$  ENTÃO
        melhsol =  $s^*$ ;
        melhiter = niter;
        Fmelhor =  $F(s^*)$ ;

    FIM SE
    s =  $s^*$ ;
FIM ENQUANTO

```

Figura 6.11 - Algoritmo BT [Glover, 1997].

Para a utilização do método BT, é fundamental a definição da função objetivo (F) do problema em questão. Após esta definição, é gerada uma solução inicial viável independentemente. Para geração da solução inicial, é fundamental que esta faça parte do conjunto de soluções possíveis do espaço amostral. Sempre que uma solução “s” é obtida, é gerado um subconjunto V^* de $N(s)$ e realizado o movimento para a melhor solução s^* em V^* . Se $N(s)$ não é muito grande é possível fazer $V^*= N(s)$ [Viana, 1998].

A utilização do critério de melhor movimento na BT é baseada na suposição de que movimentos bem avaliados têm grande probabilidade de conduzir a uma solução ótima (ou próxima do ótimo) [Gómez, 1996]. De maneira a evitar ciclos e mínimos locais, não é permitido voltar às soluções que foram visitadas em um número “k” de movimentos anteriores. Esta estratégia é chamada de lista tabu ou memória de curto prazo da BT. Esta lista de comprimento “h” é interpretada como uma fila e é chamada de lista tabu. Sempre que um movimento de “s” para s^* é executado, o movimento é introduzido no fim da lista e o movimento mais antigo é removido. No momento em que o movimento que conduz a solução “s” entra na lista, todos os movimentos que retornam à “s” ficam proibidos nas próximas “k” iterações, ou seja, “s” tornou-se lista tabu e qualquer movimento que leve a “s” é considerado não permitido [Glover, 1997; Viana, 1998].

O comprimento da lista tabu controla a memória do espaço de busca. Sendo assim, tendo uma lista pequena a busca se concentrará em pequenas áreas do espaço de busca. Caso contrário, uma lista grande força o processo de busca a explorar grandes regiões, porque ela proíbe visitar novamente um alto número de soluções. O comprimento da lista tabu pode ser variado durante a busca, utilizando assim algoritmos mais robustos.

Para Glover [Glover, 1997], é necessário que a lista tabu tenha um comprimento definido, pois poderá haver momentos em que seja necessário voltar para alguma solução, e a partir dela, buscar outras soluções. Além do comprimento da lista tabu, [Viana, 1998] afirma que o tamanho da vizinhança é outro parâmetro importante para o uso da BT. A escolha pelo tamanho da vizinhança é fator determinante para o tempo de execução do algoritmo. Uma vizinhança pobre e grande pode comprometer a execução do algoritmo; assim este valor deve ser ajustado, juntamente com o número máximo de iterações e o tamanho da lista tabu.

É fundamental entender que, ao proibir um número muito grande de soluções, o algoritmo poderá perder desempenho, devido aos ciclos repetidos de soluções. Portanto, é importante que uma determinada solução, ou atributo saia da lista tabu. Este critério de retirada do movimento é chamado de critério de aspiração [Viana, 1998]. Segundo Glover [Glover, 1997], o critério de aspiração comumente usado é a seleção de soluções que são

melhores do que a solução corrente. O uso apropriado de tal critério pode ser muito importante para capacitar o método BT a alcançar níveis de melhor performance [Reeves, 1993].

Dado que alguns movimentos considerados tabu podem não ser absolutos, é introduzido um procedimento para cancelar a situação tabu de um movimento quando este leva a uma melhoria [Gómez, 1996]. Uma função de critério de aspiração $A(z)$ é definida para cada valor “z” da função objetivo. Se a solução vizinha s^* está na lista tabu, porém $F(s^*) < A(z = F(s))$, então s^* é considerada como um solução normal de V^* e dessa forma o movimento é perdoado.

Duas regras podem ser definidas como critério de parada na BT. A primeira é pela definição de um número máximo de iterações que serão realizadas sem se obter uma solução ótima em relação a melhor solução obtida. A segunda regra condiciona o critério de parada quando a melhor solução obtida é considerada muito próxima do limite inferior de (F_{min}). Normalmente, este limite não é conhecido com suficiente precisão [Glover, 1997].

6.2.4 Aplicações

Frente ao desafio de resolver alguns problemas de otimização, a técnica BT apresenta-se bastante eficiente. Em uma variedade de cenários, tem-se encontrado soluções superiores às melhores soluções já obtidas previamente por métodos alternativos. Em Glover [Glover, 1997] são demonstradas várias áreas de aplicação da BT, sendo que podemos destacar as seguintes:

- sistemas de Manufaturra e planejamento (ex: *Flow-shop* e *Job-shop*);
- telecomunicações;
- computação paralela;
- transporte, roteamento e projetos de redes;
- grafos (particionamento de grafos, o problema da p-Mediana e o problema do clique máximo);
- redes neurais;
- otimização estocástica e contínua; e
- análise financeira.

A seguir, são apresentados alguns exemplos de aplicações da BT.

6.2.4.a Escalonamento em Sistemas de Manufatura

A utilização de técnicas de escalonamento eficiente em sistemas de manufatura, tende à redução de custos (ex: custos de inventário, custos de trabalho, etc.) e, ainda melhoram a eficiência operacional de gerenciamento. Os estudos de problemas mais frequentemente usados e encontrados na literatura para estes casos são: (a) o problema do *Job-shop* e, (b) o problema do *Flow-shop*.

Em Nowicki [Nowicki, 1996], é possível ver o desenvolvimento de métodos BT para os problemas (a) e (b), que buscam otimizar o tempo total de produção. Os algoritmos propostos empregam uma memória tabu, a qual é responsável por armazenar atributos de soluções já visitadas, que é representada por pares selecionados de serviços adjacentes em uma máquina. A intensificação e a diversificação ocorrem armazenando a melhor solução coletada durante a busca. Uma seqüência prolongada de passos não produtivos gera uma trajetória da pesquisa para a próxima melhor solução, a qual é recuperada juntamente com seu histórico, como uma base para re-iniciar a pesquisa.

6.2.4.b O problema do Roteamento de Veículos

O problema do roteamento de veículos dá-se por um conjunto de cidades, cada qual com uma demanda “x” por um produto, e um depósito com veículos de capacidade “y”. Deve-se então, encontrar as rotas para os veículos, minimizando os custos de transporte e atendendo a todas as cidades.

Gendreau [Gendreau, 1994] desenvolveu uma heurística baseada em BT, chamada de *TABUROUTE*. O algoritmo considera uma seqüência de soluções adjacentes obtidas por repetidamente remover um vértice da sua rota corrente, e re-inserindo em uma outra rota. Os resultados obtidos em comparação com outras soluções mostraram que a BT se mostrou a melhor heurística existente para o problema de roteamento de veículos, e entre as propostas de BT, *TABUROUTE* sempre produz a melhor solução conhecida. O sucesso de *TABUROUTE* pode ser atribuído à duas técnicas implementadas. Primeiramente, é permitida a pesquisa visitar soluções impraticáveis, as quais são gerenciadas por meio de uma função objetivo com termos de penalidade. Segundo, um mecanismo de inserção altamente eficaz é empregado. O procedimento de inserção tem a habilidade de perturbar periodicamente as soluções e assim, reduzir o risco de encontrar um ótimo local.

A maior vantagem do *TABUROUTE* é sua flexibilidade. O procedimento pode começar de uma solução praticável ou mesmo não-praticável. Ele pode ser adaptado ao contexto no qual a quantidade de veículos é fixa ou limitada, ou onde os veículos não são

homogêneos. Por último, o algoritmo pode facilmente tratar de características adicionais do problema como designar cidades em particular a veículos específicos, usar vários depósitos, e permitir rotas primárias e secundárias.

6.2.4.c Modelo de escalonamento de partes em uma máquina versátil

Em Gómez [Gómez, 1996] é utilizado a BT para um modelo de seqüenciamento de partes considerando uma máquina versátil. É considerada a capacidade do magazine, datas de entrega e os turnos de produção. A função objetivo baseia-se na minimização do número de *setups*, troca de ferramentas, tempo total de produção, tempo ocioso no final do turno e tempo total de atraso. Cada parcela citada possui um peso associado. O algoritmo desenvolvido utiliza duas dimensões, sendo a primeira em função do tempo e a outra em função da seleção de partes.

6.2.4.d Auditoria de Escalonamento de produção

Dodin *apud* [Glover, 1997] desenvolveu um procedimento que utiliza as regras de despachos tradicionais com as estruturas de memória de curto e longo prazo da BT. A BT intensifica a procura usando a memória de curto prazo e diversifica a busca com as regras de despacho controladas, memória de longo prazo e listas candidatas. A abordagem através da BT produziu resultados superiores aos procedimentos heurísticos tradicionais.

7 Modelo Proposto

Neste capítulo é apresentado o modelo aplicado ao JSSP num SMF. Para tanto, é mostrado na seção 7.1 o ambiente de SMF na qual o modelo é implementado. Já na seção 7.2 é contextualizada a formulação do modelo e na seção 7.3 é apresentada a arquitetura.

7.1 Ambiente do Sistema de Manufatura Flexível

A partir da contextualização dos Sistemas de Manufatura (capítulo 2) e dos Sistemas de Manufatura Flexível (capítulo 4), o ambiente do SMF na qual o modelo será implementado, é ilustrado na Figura 7.1.

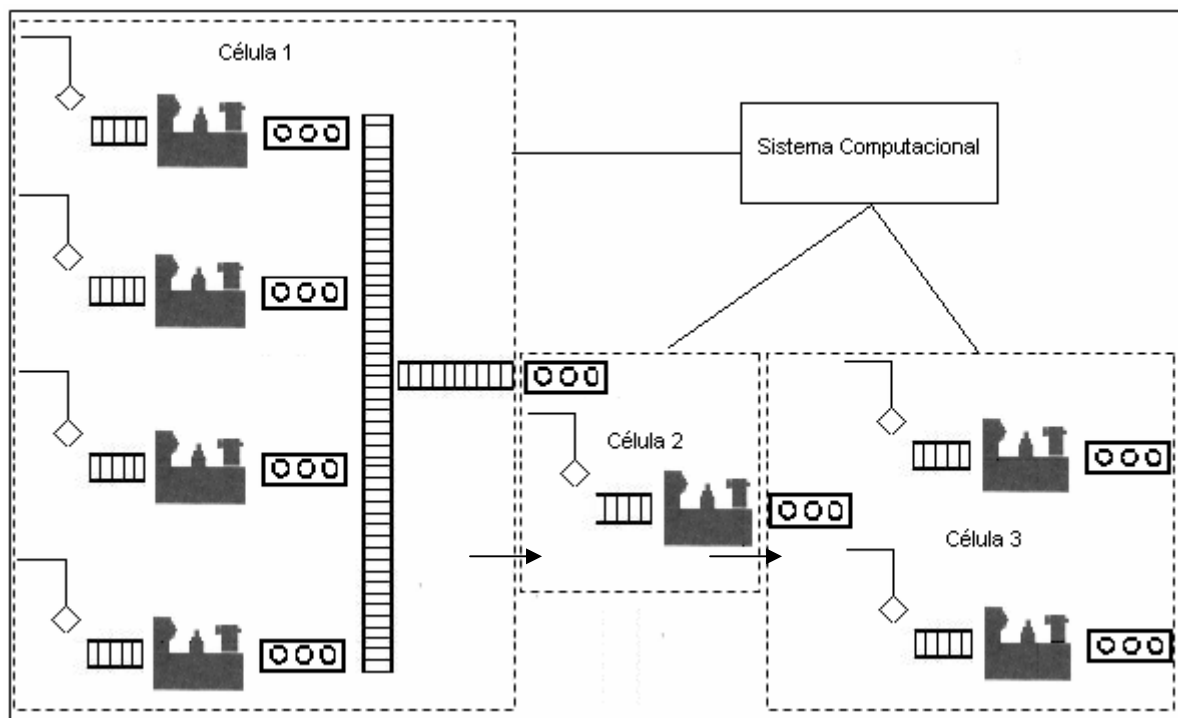


Figura 7.1 - Ambiente do SMF.

O ambiente do SMF apresenta uma planta de produção do ramo metal-mecânico. Esta planta está dividida em três células de manufatura controladas por um sistema computacional, sendo que:

- primeira célula de manufatura: é a primeira fase de produção e é denominada de estamparia de aço, local este em que a matéria-prima transforma-se em parte. Esta

célula de manufatura é composta por quatro máquinas do tipo prensa hidráulica. Para cada máquina, existe um robô que realiza o manuseio (transporte) do material para uma esteira que por sua vez conecta diretamente com a máquina a fim de realizar a produção. Após a passagem nesta primeira fase, a parte fica temporariamente num *buffer*⁵ para a entrada na próxima célula de manufatura. A Figura 7.2 apresenta a definição da simbologia utilizada na planta;



Figura 7.2 - (a) Robô (b) Esteira de transporte (c) Máquina (d) *Buffer* temporário.

- segunda célula de manufatura: após a realização da primeira fase de produção, através de um robô, é feito o carregamento da parte semi-pronta para a segunda fase de produção que é denominada têmpera. Nesta fase o produto adquire a dureza. A máquina utilizada nesta fase é chamada de Forno Têmpera; e
- terceira célula de manufatura: Adquirida a dureza pela parte (fase anterior), esta é descarregada num *Buffer* para a entrada na fase de embalagem da parte para a entrega ao cliente. Nesta Célula de Manufatura é utilizado um robô para o transporte da parte até a esteira. As máquinas utilizadas nesta fase são máquinas semi-automatizadas, sendo necessários trabalhadores para auxílio na organização da parte dentro da embalagem.

O sistema computacional utilizado neste ambiente está dividido em cinco níveis de controle conforme apresentado na Figura 7.3.

⁵ É o termo utilizado para um local de armazenamento temporário da parte na produção [Jha, 1991; Lorini, 1993; Kusiak, 1994].

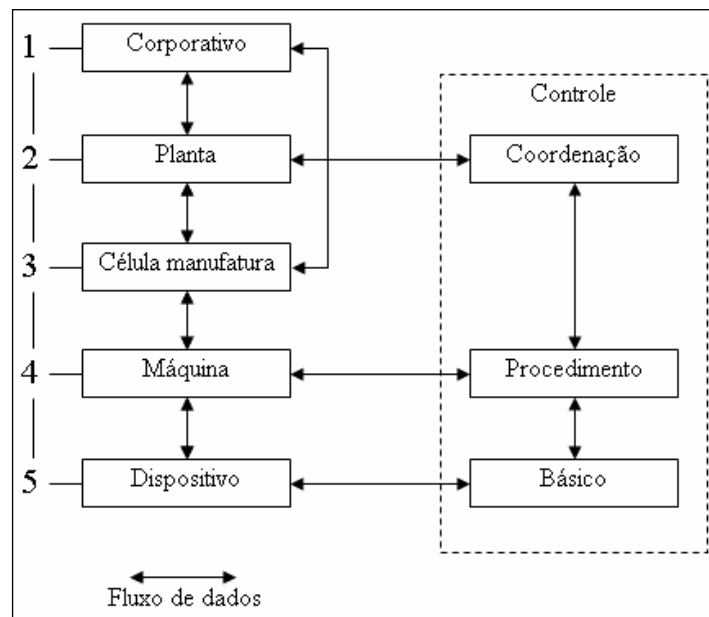


Figura 7.3 - Níveis de controle [Groover, 2001].

7.2 Formulação do Modelo

Após a definição do ambiente de SMF, esta seção tem como objetivo apresentar a formulação aplicada ao JSSP. Está baseado nos modelos propostos por Widmer, Gen, Gómez e Blazewicz [Widmer, 1991; Gen, 1996; Gómez, 1996; Blazewicz, 2002]. A formulação do escalonamento das partes para o JSSP é mostrada a seguir.

Sejam:

m = número de máquinas;

n = número de partes;

e = escalonamento;

p = FP;

i = índice para a parte;

j = parte que é processada após a parte i ;

k = índice para a máquina;

h = máquina que precede a máquina k ;

De_i = data de entrega da parte i ;

Dsp_{ik} = data da saída de produção para a parte i na máquina k ;

C_{ik} = tempo inicial da parte i na máquina k ;

T_{ik} = tempo de processamento da parte i na máquina k ;

M = número positivo maior que o tempo total de processamento para n ;

$$A_{ihk} = \begin{cases} 1, & \text{se o processamento na máquina h precede à máquina k para a parte i} \\ 0, & \text{caso contrário} \end{cases}$$

$$X_{ijk} = \begin{cases} 1, & \text{se a parte i precede a parte j na máquina k} \\ 0, & \text{caso contrário} \end{cases}$$

$$P_{ik} = \begin{cases} 1, & \text{se a parte i possui operação na máquina k} \\ 0, & \text{caso contrário} \end{cases}$$

Função objetivo:

Minimizar

$$f(e, p) = p_1 \cdot makespan(e, p) + p_2 \cdot atraso(e, p) + p_3 \cdot parada(e, p) + p_4 \cdot ociosidade(e, p) \quad (7.1)$$

Onde:

$$makespan(e, p) = \sum_{k=1}^m \sum_{i=1}^n makespan_{ik}, \text{ tal que } makespan_{ik} > 0, \text{ para } i = 1, 2, \dots, n, \quad (7.2)$$

$$k = 1, 2, \dots, m;$$

$$atraso(e, p) = \sum_{k=1}^m \sum_{i=1}^n (Dsp_{ik} - De_i), \text{ tal que } (Dsp_{ik} - De_i) > 0 \text{ e } P_{ik} = 1, \quad (7.3)$$

$$\text{para } i = 1, 2, \dots, n,$$

$$k = 1, 2, \dots, m;$$

$$parada(e, p) = \sum_{k=1}^m \sum_{i=1}^n parada_{ik}, \text{ tal que } parada_{ik} > 0, i = 1, 2, \dots, n, k = 1, 2, \dots, m; \quad (7.4)$$

$$ociosidade(e, p) = \sum_{k=1}^m \sum_{i=1}^n ociosidade_{ik}, \text{ tal que } ociosidade_{ik} > 0, i = 1, 2, \dots, n, \quad (7.5)$$

$$k = 1, 2, \dots, m;$$

$$C_{ik} - T_{ik} + M(1 - A_{ihk}) \geq C_{ih}, \text{ tal que } i = 1, 2, \dots, n, \quad (7.6)$$

$$h, k = 1, 2, \dots, m;$$

$$C_{jk} - C_{ik} + M(1 - X_{ijk}) \geq T_{jk}, \text{ tal que } i, j = 1, 2, \dots, n, \quad (7.7)$$

$$k = 1, 2, \dots, m;$$

$$C_{ik} \geq 0, \text{ tal que } i = 1, 2, \dots, n, \quad (7.8)$$

$$k = 1, 2, \dots, m;$$

$$X_{ijk} \in \{0, 1\}, \text{ tal que } i, j = 1, 2, \dots, n, \quad (7.9)$$

$$k = 1, 2, \dots, m; e$$

$$p_1, p_2, p_3, p_4 \geq 0. \quad (7.10)$$

A função objetivo (7.1) busca a minimização e é definida pela variável “e” (escalonamento), que representa a dimensão temporal e a variável “p” (Famílias de Partes), representa a dimensão física para o modelo proposto. É formada por quatro variáveis de decisão que refletem uma estratégia de otimização. A variável de decisão *makespan*(e,p) representa o tempo total de produção, ou seja, é o tempo inicial da primeira parte processada em produção até o tempo final da última parte processada. Já a variável de decisão atraso(e, p) representa o tempo total de atraso. Já o tempo total de parada é representada pela variável de decisão parada(e,p) e significa o tempo entre dois lotes de produção. A variável de decisão ociosidade(e,p) representa o tempo não utilizado dos turnos.

A restrição (7.2) assegura o somatório total do tempo de produção. Já a restrição (7.3) garante o somatório total do tempo de atraso através da data de entrega da parte e a data de saída desta. A restrição (7.4) assegura o tempo total de paradas e a restrição (7.5) assegura o somatório do tempo total ocioso. A restrição (7.6) garante que a seqüência das operações (rota) para cada parte seja respeitada e, a restrição (7.7) assegura que cada máquina processa somente uma parte por vez. As restrições (7.8) e (7.9) garantem o limite inferior e superior para “i,j e k” e, por fim, a restrição (7.10) assegura a não-negatividade.

Através dos valores definidos para os pesos (p_1, p_2, p_3, p_4) de cada variável de decisão, a função objetivo pode refletir as seguintes estratégias de otimização: (a) minimizar o tempo total de produção, (b) minimizar o tempo total de atrasos, (c) minimizar o tempo total de paradas e (d) minimizar o tempo total não utilizado [Gómez, 1996]. Além das suposições

mencionadas no JSSP, também são respeitados os tempos de processamento para cada turno e as datas de entrega.

Todas as restrições consideram as máquinas do ambiente de SMF representada pelo índice “m” e o número total de partes representada pelo índice “n”. A unidade de tempo utilizada está em minutos.

7.3 Arquitetura do Modelo

A arquitetura para a Seleção de Partes e Escalonamento está dividida em cinco fases de aplicação conforme é apresentado na Figura 7.4.

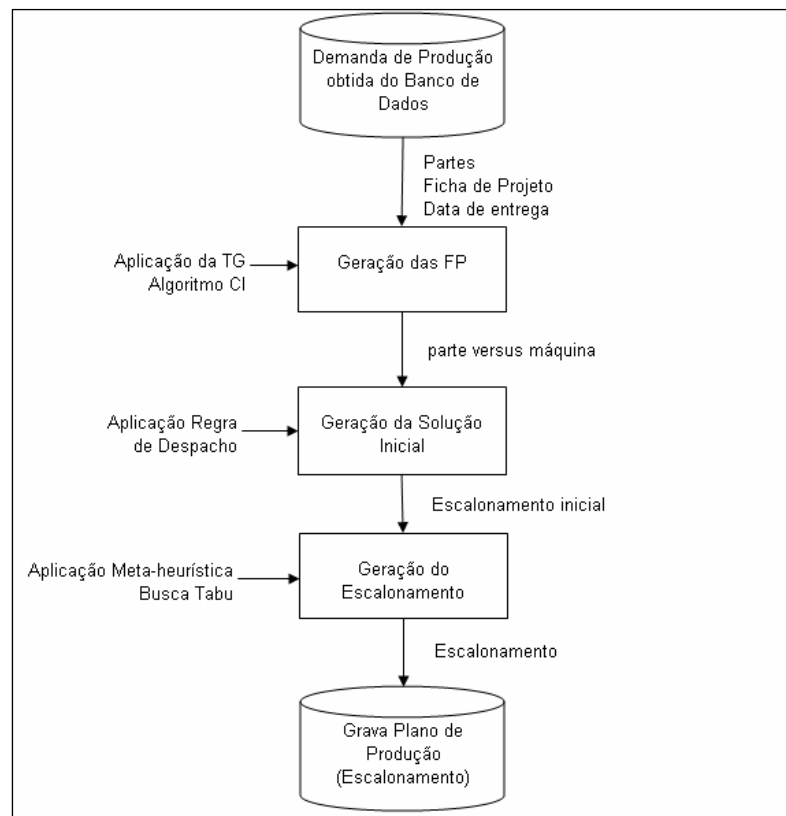


Figura 7.4 - Arquitetura do modelo.

A primeira fase é responsável por obter a demanda de produção através das informações contidas no Banco de dados, sendo recuperadas as informações das partes (ficha de projeto) e data de entrega. Já na segunda fase, que utiliza a TG, realiza a geração da matriz parte versus máquina aplicando o algoritmo CI para identificação de agrupamentos. Na terceira fase, é utilizado o algoritmo de regras de despacho para a geração do escalonamento inicial. Em Kusiak [Kusiak, 1992] são denominadas regras de despacho aqueles algoritmos em que um escalonamento inicial é obtido através de regras relacionadas às partes e

máquinas. O escalonamento inicial deve ser viável, ou seja, deve ser uma solução que esteja contida no espaço amostral das soluções possíveis para o problema em questão.

A partir do escalonamento inicial, é possível realizar a quarta fase que consiste na aplicação da meta-heurística BT para a obtenção do escalonamento final. Após a geração do escalonamento final é gravado o escalonamento final em planos de produção, possibilitando assim observar seus históricos e compará-los com o escalonamento efetivo.

8 Aplicação

Neste capítulo são descritas as cinco fases da arquitetura do modelo, de forma a entender a entrada, processamento e saída de cada fase. De forma a entender a aplicação, segue abaixo, o diagrama de classes utilizado no modelo, onde: a classe *Plan* é responsável pela primeira, segunda e quinta fase do modelo, a classe *DispatchingRules* realiza a terceira fase do modelo e, por fim, a classe *TabuSearch* é responsável pela otimização do escalonamento.

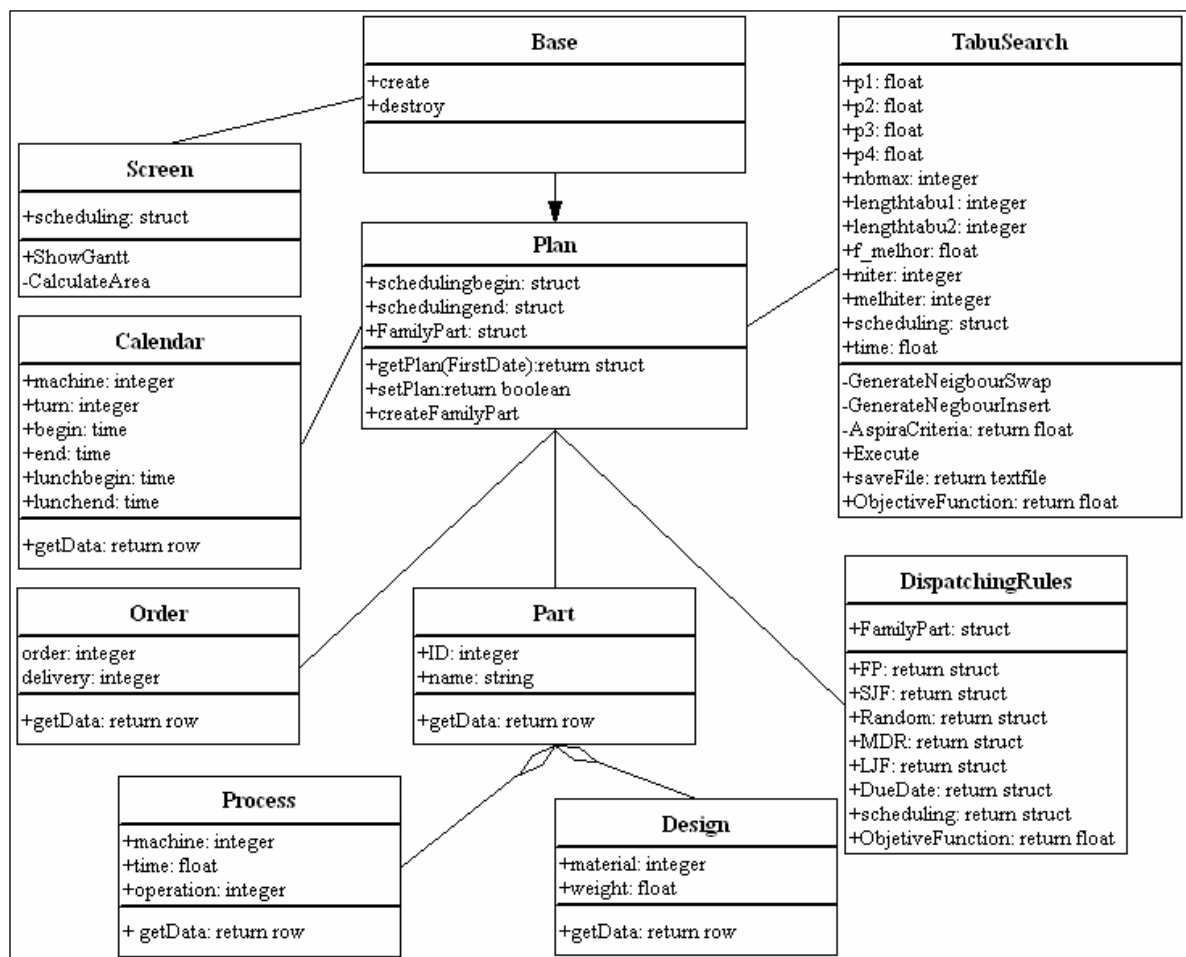


Figura 8.1 - Diagrama de Classes.

O tipo de dado *struct* corresponde a uma tabela de memória que possui métodos de criação dinâmica de colunas, filtros, impressão, localização e ordenação de dados. Dessa forma, é possível realizar o acesso direto aos dados sem haver necessidade de procurar em toda a tabela como é realizado em matrizes. A seguir são detalhadas as fases do modelo.

8.1.1 Demanda de Produção

Nesta fase é obtida a demanda de produção juntamente com as informações da ficha de projeto e a data de entrega prevista da parte. As reprogramações também são realizadas nesta fase, sendo que para a reprogramação também é gerado o plano de produção. A Tabela 8.1 ilustra as informações obtidas após a primeira fase do modelo.

Tabela 8.1 - Informações de produção após a primeira fase da arquitetura

Parte	Descrição parte	Qtde	Data entrega	Máquina							Rota									
				226	265	301	312	400	501	515	226	265	301	312	400	501	515			
1	Alma 12X00X1,35 CT	7000	21/2/2006	1,2		2,3						2		1						
2	Alma 12X00X1,10 CT	2500	21/2/2006		1,2		3,5			1,9		2			1					3
3	Alma 12X00X1,10 CT	2000	21/2/2006		1,4		3,2			1,6		1			2					3
4	Alma 12X00X1,35 ST	3500	21/2/2006	1,1		2,6						2		1						
5	Alma 11X00X1,35 CT	2880	21/2/2006					3,7	5,3							1			2	
6	Alma 11X00X1,35 CT	4400	21/2/2006					2,7	5							1			2	
7	Alma 12X00X1,25 CT	6570	21/2/2006	1,5		2,7						1		2						
8	Alma 12X00X1,10 ST	4320	21/2/2006		1,3		3,1						3		2					1
9	Alma 11X00X1,35 CT	3760	21/2/2006					3,2	5,2							2			1	
10	Alma 12X00X1,35 CT	4400	21/2/2006	1,6		2,7						1		2						

As máquinas e rotas de produção são conhecidas e extraídas da ficha de projeto e não existe restrição quanto à capacidade das máquinas. A data de entrega é proveniente da necessidade do cliente.

Verificando a primeira linha da tabela, obtêm-se as seguintes informações: a parte 1 tem uma demanda de produção de 7000 unidades e deverá ser entregue no dia 21/2/2006. Esta parte é processada nas máquinas “301” e “226”, com um tempo de processamento de 2,3 e 1,2 segundos, respectivamente. A coluna rota define a ordem de processamento da parte nas máquinas. Cabe lembrar que, as informações contidas nesta tabela referem-se às partes sem plano de produção, ou seja, não-escalonadas.

8.1.2 Geração das Famílias de Partes

A fase dois da arquitetura consiste na seleção de partes de modo a agrupá-las através de similaridades, formando assim as Famílias de Partes. É utilizado o algoritmo CI proposto por Kusiak [Kusiak, 1992]. A Tabela 8.2 ilustra o agrupamento.

Tabela 8.2 - Agrupamento das partes após a segunda fase da arquitetura

Parte	Descrição parte	Qtde	Data entrega	Máquina							Rota						Famílias de Partes								
				226	265	301	312	400	501	515	226	265	301	312	400	501	515	226	265	301	312	400	501	515	
1	Alma 12X00X1,35 CT	7000	21/2/2006	1,2		2,3								2		1								2	
2	Alma 12X00X1,10 CT	2500	21/2/2006		1,2		3,5			1,9				2	2		1						3		2
3	Alma 12X00X1,10 CT	2000	21/2/2006		1,4		3,2			1,6				1		2							3		2
4	Alma 12X00X1,35 ST	3500	21/2/2006	1,1		2,6								2		1							2		
5	Alma 11X00X1,35 CT	2880	21/2/2006					3,7	5,3									1	2						
6	Alma 11X00X1,35 CT	4400	21/2/2006					2,7	5									1	2						
7	Alma 12X00X1,25 CT	6570	21/2/2006	1,5		2,7					1		2										1		
8	Alma 12X00X1,10 ST	4320	21/2/2006		1,3		3,1			1,8				3		2									
9	Alma 11X00X1,35 CT	3760	21/2/2006					3,2	5,2									2	1						1
10	Alma 12X00X1,35 CT	4400	21/2/2006	1,6		2,7					1			2									1		

O algoritmo para agrupamento das partes utiliza a coluna “rota” como referência para a geração da FP, resultando assim na coluna “Famílias de Partes” para cada máquina. O critério para a escolha da parte que irá gerar a FP deste modelo é a escolha da parte com máquinas similares e operações a serem processados. A Figura 8.2 mostra o algoritmo para a geração de FP.

```

início
carregar tabela parte versus rota para A;
inicializar tabela P com a coluna rota de A;
contador de FP k := 1;

enquanto P existe entrada diferente de "0" faça
  se não existir maquina "j" compartilhada com P(k) então
    gerar nova tabela P(k + 1) somente com entradas
    sem compartilhamento;
    k := 1;
  fim se;
  se existir uma maquina "j" compartilhada com P(k) então
    se a seqüência operacional de "j" for diferente da
    seqüência operacional P(k) então
      k := k + 1;
    marcar esta máquina em P(k)
  fim se;
  atualizar A com P(k);
  atualizar tabela P(k);
fim enquanto;

```

Figura 8.2 - Algoritmo para a geração de FP.

Basicamente o algoritmo gera uma tabela “P” com base na coluna “Rota” da Tabela 8.1. O algoritmo executará até que não exista entrada diferente de “0” na tabela, e a geração da FP é realizada através da variável “k”, dado que a operação de processamento da parte é igual, caso contrário, é incrementado “k”. Quando não existir máquinas compartilhadas, é gerada uma nova matriz “P” com a iteração “k = 1”, considerando somente as entradas

diferentes de “0”. Nesta fase foi acrescentada a coluna “Famílias de Partes” que, até o momento, era inexistente.

8.1.3 Geração da Solução Inicial

A geração do escalonamento inicial que é a terceira fase da arquitetura baseia-se na utilização de regras de despacho. Em Hoffmann [Hoffmann, 2001] são consideradas as seguintes regras de despacho:

- regra randômica (RR): as partes são seqüenciadas aleatoriamente;
- recursos mais dissimilares (RMD): o seqüenciamento é realizado de forma que uma parte compartilhe o menor número de recursos com a parte seguinte no seqüenciamento;
- famílias de Partes (FP): as partes são seqüenciadas de forma crescente segundo o número de FP à qual pertencem;
- processos mais curtos primeiro (PMCP): a partir do tempo de processamento para cada parte, estas são seqüenciadas de forma crescente;
- processos mais longos primeiro (PMLP): a partir do tempo de processamento para cada parte, estas são seqüenciadas de forma decrescente;
- recursos de famílias mais similares (RFMS): o seqüenciamento é gerado tal que uma parte compartilhe o maior número de recursos com a próxima parte no escalonamento;
- datas de entrega mais recente (DEMR): as partes que possuem a data de entrega mais recente, são as primeiras a serem seqüenciadas; e
- datas de entrega mais longa (DEML): as partes que possuem o maior prazo de entrega são as primeiras a serem seqüenciadas.

A partir da regra de despacho selecionada, o objetivo é a geração de uma solução inicial que seja viável, ou seja, faça parte do espaço amostral das soluções possíveis. O algoritmo selecionado deverá seguir os seguintes passos: (a) carregar os dados da Tabela 8.1, (b) carregar o calendário fabril e (c) gerar o escalonamento inicial de acordo com a regra de despacho selecionada. Para exemplificar, o calendário fabril é configurado para 7 máquinas e, para cada máquina, é definido os dias de trabalho, formando assim a capacidade fabril; cada dia de trabalho está dividido em três turnos de 480 minutos. Neste exemplo, foi considerado também, o tempo de parada para troca de material (10 minutos), o tempo para troca de ferramentas (8 minutos) e o tempo para troca de tamanho da parte (5 minutos). A Figura 8.3

apresenta o escalonamento inicial utilizando a regra de despacho FP que obteve o *makespan* de 1371 minutos. A regra de despacho FP foi escolhida, pois é possível agrupar e gerar lotes por similaridades, gerar as vizinhanças do algoritmo BT através de movimentos de troca de dois lotes e de retirada e inserção de uma parte em outro lote, e ainda a utilização de outra regra de despacho para cada FP gerada, como por exemplo, a regra DEMR

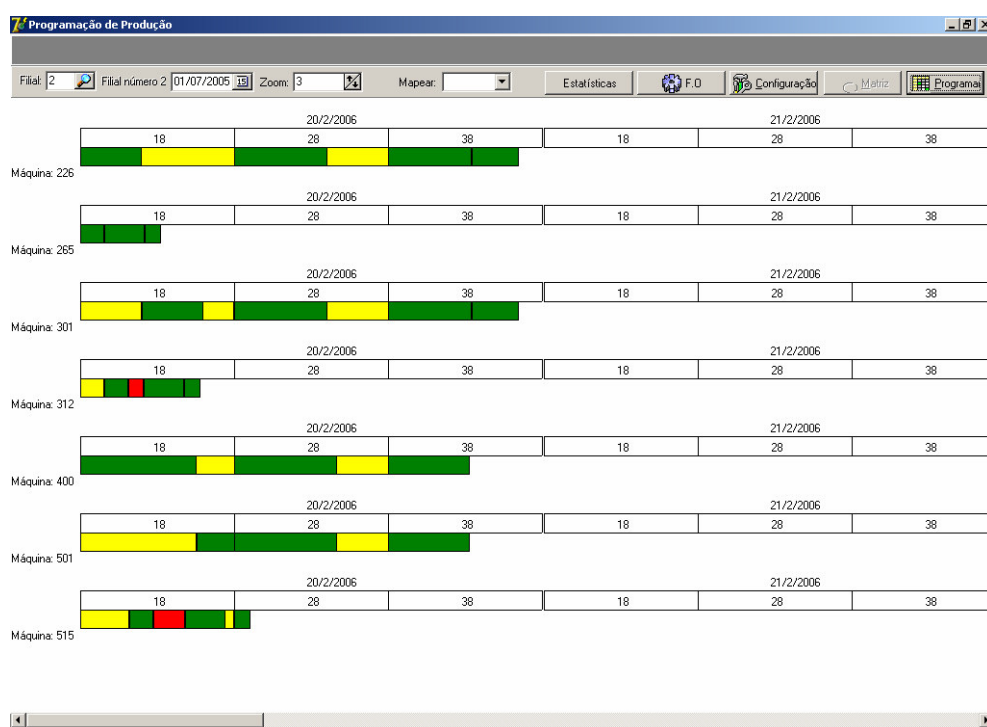


Figura 8.3 - Escalonamento inicial após a terceira fase da arquitetura.

A cor vermelha significa que ocorreu uma parada de produção em virtude da troca de lotes⁶. Já a cor amarela, indica que naquele momento a máquina não está trabalhando, ou seja, está ociosa. A Tabela 8.3 ilustra a nova configuração após a execução da terceira fase.

Tabela 8.3 - Configuração após a aplicação da regra de despacho

Parte	Descrição parte	Qtde	Data entrega	Saída Produção	Máquina							Rota							Famílias de Partes						
					226	265	301	312	400	501	515	226	265	301	312	400	501	515	226	265	301	312	400	501	515
1	Alma 12X00X1,35 CT	7000	21/2/2006	20/2/2006	1,2		2,3							2		1				2		1			
2	Alma 12X00X1,10 CT	2500	21/2/2006	20/2/2006		1,2		3,5			1,9			2		1				3		2		1	2
3	Alma 12X00X1,10 CT	2000	21/2/2006	20/2/2006		1,4		3,2			1,6			1		2				3		1		2	2
4	Alma 12X00X1,35 ST	3500	21/2/2006	20/2/2006	1,1		2,6						2		1					2		1			
5	Alma 11X00X1,35 CT	2880	21/2/2006	20/2/2006					3,7	5,3							1	2					1	2	
6	Alma 11X00X1,35 CT	4400	21/2/2006	20/2/2006					2,7	5						1	2						1	2	
7	Alma 12X00X1,25 CT	6570	21/2/2006	20/2/2006	1,5		2,7						1		2					1		2			
8	Alma 12X00X1,10 ST	4320	21/2/2006	20/2/2006		1,3		3,1			1,8			3		2				1		3		2	1
9	Alma 11X00X1,35 CT	3760	21/2/2006	20/2/2006					3,2	5,2						2	1						2	1	
10	Alma 12X00X1,35 CT	4400	21/2/2006	20/2/2006	1,6		2,7						1		2					1		2			

⁶ A formação do lote ocorre através da junção da FP, data de fabricação, turno e máquina. Ex: Lote nro: 1-20-2-2006-18-226. Este lote é válido para as partes p7 e p10 da Figura 45.

É possível verificar que o atributo “Saída Produção” foi adicionado após a execução da terceira fase. Através dos atributos “Data de Entrega” e “Saída Produção” é analisado o atraso, neste caso, não ocorreram atrasos.

Para a realização da terceira fase, outras duas estruturas de controle são criadas de forma a garantir a integridade do modelo. A Tabela 8.4 ilustra o seqüenciamento das partes na máquina e, na Tabela 8.5, é apresentado o calendário fabril para a máquina “515” com os valores para cada variável de decisão. Para ilustração do calendário fabril foram desconsideradas as demais máquinas em virtude do volume de dados.

Tabela 8.4 - Seqüenciamento das partes na máquina após terceira fase da arquitetura

Máquina	A partir de	Seqüência			
		1	2	3	4
226	20/2/2006	p7	p10	p1	p4
265	20/2/2006	p3	p2	p8	
301	20/2/2006	p1	p4	p7	p10
312	20/2/2006	p2	p3	p8	
400	20/2/2006	p5	p6	p9	
501	20/2/2006	p9	p5	p6	
515	20/2/2006	p8	p2	p3	

Tabela 8.5 - Calendário fabril da máquina “515” após terceira fase da arquitetura

Máquina	Dia	Turno	<i>makespan</i>	Tempo parado	Instantes parada	Tempo atraso	Tempo ocioso
515	20/2/2006	18	480	18	2	0	95
515	20/2/2006	28	532	0	0	0	0
515	20/2/2006	38	0	0	0	0	0

Através das informações da tabela acima, pode-se extrair as seguintes informações: no dia 20/2/2006 e no turno 18 (primeira linha da tabela), constatam-se dois instantes de parada, onde o primeiro refere-se aos 10 minutos para troca de material e os 8 minutos restantes, à troca de ferramenta. Observa-se também que, há ocorrência de tempo ocioso.

8.1.4 Geração do Escalonamento

Após o escalonamento inicial, obtido através da regra de despacho FP, é realizada a geração do escalonamento final de modo a refletir uma estratégia de otimização. Para isso, é utilizado o algoritmo Busca Tabu que, através da função objetivo descrita no modelo, realiza a

otimização do escalonamento. A Figura 8.4 mostra o escalonamento após a quarta fase do modelo.

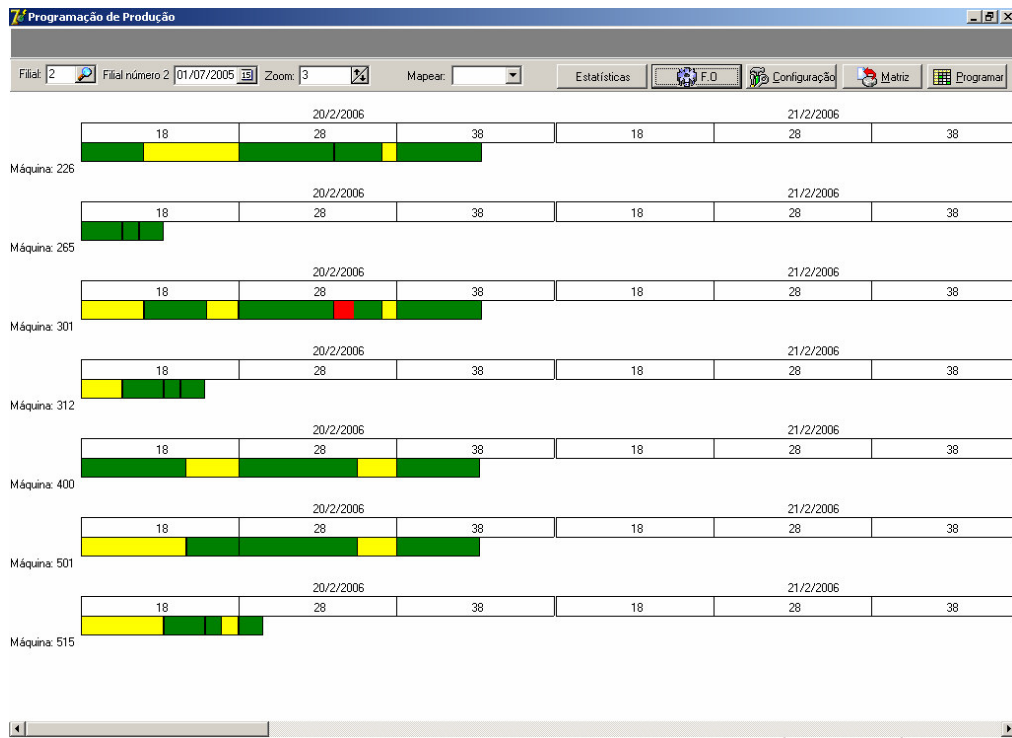


Figura 8.4 - Escalonamento final após a quarta fase da arquitetura.

Conforme ilustrado acima, o escalonamento final obteve o *makespan* de 1222 minutos. A Tabela 8.6 mostra a nova configuração após a execução do algoritmo.

Tabela 8.6 - Configuração após a aplicação do algoritmo BT

Parte	Descrição parte	Qtde	Data entrega	Saída Produção	Máquina						Rota						Famílias de Partes								
					226	265	301	312	400	501	515	226	265	301	312	400	501	515	226	265	301	312	400	501	515
1	Alma 12X00X1,35 CT	7000	21/2/2006	20/2/2006	1,2		2,3						2		1				2		1				
2	Alma 12X00X1,10 CT	2500	21/2/2006	20/2/2006		1,2		3,5					2	2					3	2		1			2
3	Alma 12X00X1,10 CT	2000	21/2/2006	20/2/2006		1,4		3,2					1		2				3	1		2			2
4	Alma 12X00X1,35 ST	3500	21/2/2006	20/2/2006	1,1		2,6						2		1					2					
5	Alma 11X00X1,35 CT	2880	21/2/2006	20/2/2006					3,7	5,3						1	2						1	2	
6	Alma 11X00X1,35 CT	4400	21/2/2006	20/2/2006					2,7	5						1	2						1	2	
7	Alma 12X00X1,25 CT	6570	21/2/2006	20/2/2006	1,5		2,7						1		2					1		2			
8	Alma 12X00X1,10 ST	4320	21/2/2006	20/2/2006		1,3		3,1						3		2				3		2			1
9	Alma 11X00X1,35 CT	3760	21/2/2006	20/2/2006					3,2	5,2						2							2	1	
10	Alma 12X00X1,35 CT	4400	21/2/2006	20/2/2006	1,6		2,7						1		2					1		2			

Na tabela acima, embora não tenha ocorrido a mudança da data de produção, nesta fase é realizada a atualização do atributo “Saída Produção”.

Além da Tabela 8.6, são atualizadas após a aplicação da fase quatro da arquitetura, o seqüenciamento das partes por máquina, ilustrado na Tabela 8.7, e o calendário fabril para a máquina “515”, apresentado na Tabela 8.8.

Tabela 8.7 - Seqüenciamento das partes na máquina após quarta fase da arquitetura

Máquina	A partir de	Seqüência			
		1	2	3	4
226	20/2/2006	p10	p7	p4	p1
265	20/2/2006	p8	p3	p2	
301	20/2/2006	p10	p7	p4	p1
312	20/2/2006	p8	p3	p2	
400	20/2/2006	p9	p6	p5	
501	20/2/2006	p9	p6	p5	
515	20/2/2006	p8	p3	p2	

Tabela 8.8 - Calendário fabril da máquina “515” após quarta fase da arquitetura

Máquina	Dia	Turno	<i>makespan</i>	Tempo parado	Instantes parada	Tempo atraso	Tempo ocioso
515	20/2/2006	18	480	0	0	0	120
515	20/2/2006	28	557	0	0	0	0
515	20/2/2006	38	0	0	0	0	0

Embora para a máquina “515” tenha aumentado o *makespan* em relação ao escalonamento inicial, o importante é a otimização do conjunto de máquinas.

8.1.4.a Algoritmo Busca Tabu

A seguir, são apresentadas as configurações utilizadas no algoritmo BT, sendo abordado o algoritmo, o critério para a geração de vizinhanças, as políticas de geração de vizinhança, a lista tabu, o critério de aspiração e o número de iterações. A Figura 8.5 apresenta o algoritmo Busca Tabu.

```

início
e = horário inicial;
p = seleção de partes inicial;
m=máquinas;
f_melhor = f(e,p);
niter = 0;
melhiter = 0;
nbmax = número máximo de iterações sem obter uma melhora na solução;
Inicializar as listas Tabu para as duas pesquisas,  $L_e$  e  $L_p$ ;
Inicializar a função do Critério de Aspiração,  $A(z)$ ;
enquanto ( niter - melhiter < nbmax) faça
  para cada máquina “m” faça
    f' = f_melhor;
    niter = melhiter + 1;
    p = p* ;
    Gerar a partir de “m” um conjunto  $V_e^*$  de soluções  $(e,p^*)_i$ 
    em  $N_e(e,p^*)$  que não são tabu ou  $f((e,p^*)_i) < A(f(e,p^*))$ ;
    Escolher a melhor solução  $(e',p^*)$  em  $V_e^*$ ;
    Atualizar em “m” a Lista Tabu  $L_e$ , e a função do Critério de Aspiração  $A(z)$ ;
    se  $f(e',p^*) < f_melhor$ 
      f_melhor =  $f(e',p^*)$ ;
    fim se;
    e = e';

    Gerar a partir de “m” um conjunto  $V_p^*$  de soluções  $(e,p)_i$ 
    em  $N_p(e,p)$  que não são tabu ou  $f((e,p)_i) < A(f(e,p))$ ;
    Escolher a melhor solução  $(e',p')$  em  $V_p^*$ ;
    Atualizar em “m” a Lista Tabu  $L_p$ , e a função do Critério de Aspiração  $A(z)$ ;
    se  $f(e',p') < f_melhor$ 
      f_melhor =  $f(e',p')$ ;
    fim se;
    p = p';
    e = e';
  fim para;
  se f_melhor < f'
    melhiter = niter;
  fim se;
fim enquanto;

```

Figura 8.5 - Algoritmo Busca Tabu para o modelo proposto [Gómez, 1996].

Para o algoritmo ilustrado na Figura 8.5, é realizada a primeira busca através do posicionamento de dois lotes a partir do movimento de troca entre os lotes. Nesta busca, através da máquina selecionada, a vizinhança $N_e(e, p^*)$ é formada por todas as seqüências que podem ser obtidas pela troca de posição de dois lotes pertencentes aos turnos de produção. O

conjunto V_e^* de soluções $(e, p^*)_i$ é considerado igual a $N_e(h, p^*)$, ou seja, toda a vizinhança é considerada $N_e(e, p^*) = V_e^*$. Melhorando a definição de lote, pode-se agora defini-lo como um conjunto de partes que estão encadeadas em um mesmo turno e processadas por uma mesma máquina. Na realização desta primeira busca, a designação das partes às FP permanece constante (p^*).

Na segunda Busca, uma parte passa a ser processada em outro lote de mesma FP por movimentos de retirada e inserção da parte. Neste caso, a vizinhança $N_p(e, p)$ é formada por todas as seqüências possíveis que podem ser obtidas pela retirada de uma parte pertencente a um lote e por sua inserção em outro lote na mesma máquina. Analogamente, a pesquisa anterior, V_p^* é igual a $N_p(e, p)$. A parte poderá ser processada em outro turno e dia da mesma máquina, possibilitando assim a redução do tempo de atraso.

Em ambas as buscas, é realizado o procedimento de re-escalonamento dos tempos de saída das partes, de forma a calcular as paradas resultantes deste escalonamento e os tempos ociosos dos turnos.

8.1.4.a.1 Geração de Vizinhança

Conforme apresentado na Figura 8.5, o algoritmo BT apresenta duas gerações de vizinhança. A primeira geração de vizinhança baseia-se na troca de lotes, ou seja, o movimento de troca das posições de dois lotes. Já a segunda vizinhança é gerada a partir da vizinhança anterior, porém, com o movimento de retirada e inserção de partes.

No critério para a geração da primeira vizinhança que por sua vez é definida por $N_e(e, p^*)$, utiliza-se a troca de todos os lotes⁷ possíveis dentro de uma máquina, neste caso, toda a vizinhança é considerada. Nesta geração de vizinhança, as partes permanecem constantes e são representadas por p^* .

Na segunda geração de vizinhança, baseada na busca tabu anterior, uma parte pode ser processada em outro lote através do movimento de retirada e inserção da parte. A vizinhança, representada por $N_p(e, p)$, é gerada através da remoção de uma determinada parte de um lote e inserindo-a em outro lote na mesma máquina.

8.1.4.a.2 Movimento de troca de lotes

A realização do movimento através da troca de dois lotes reflete na redução do tempo total de paradas e no *makespan* dado que possa ocorrer união de lotes de mesma FP, eliminando assim

⁷ É uma seqüência de partes que possuem o mesmo processo de produção em uma determinada máquina, dentro de um mesmo turno e sem ocorrência de paradas de produção.

o tempo de parada entre os dois lotes. Segundo Gómez [Gómez, 1996], o movimento de troca de dois lotes ocorre em três etapas: (a) retirada dos dois lotes da seqüência, (b) inserção dos lotes em posições inversas em que estavam anteriormente e (c) redefinição dos lotes. Na etapa de retirada dos dois lotes, a seqüência dos demais lotes não é alterada. Os dois lotes deixam de existir nesta etapa e os tempos de processamento das demais partes ficam na espera da conclusão do movimento. A Figura 8.6 ilustra esta primeira etapa.

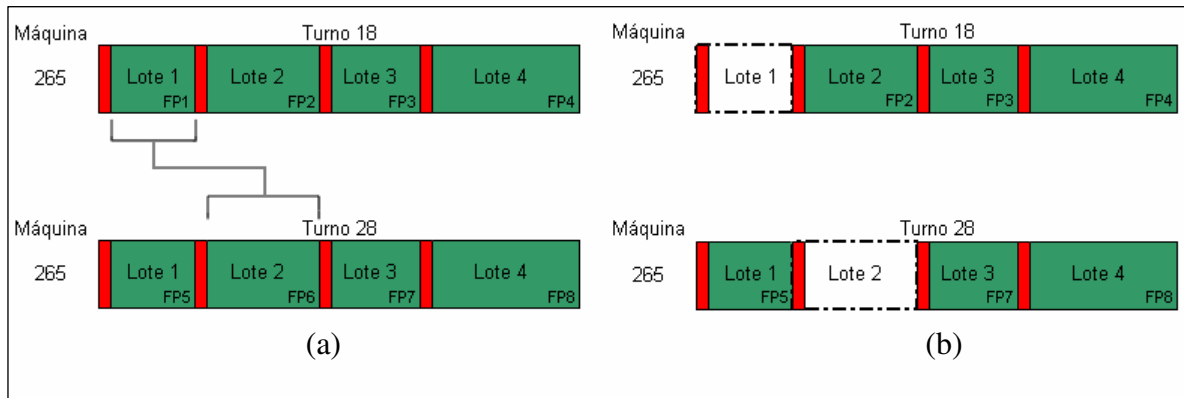


Figura 8.6 - (a) Lotes candidatos (b) Lotes retirados (as posições continuam).

Para a etapa de inserção dos dois lotes, ambos são colocados nas posições trocadas sem existir a preocupação de ultrapassagem do tempo total do turno. A Figura 8.7 apresenta o movimento de inserção dos dois lotes.

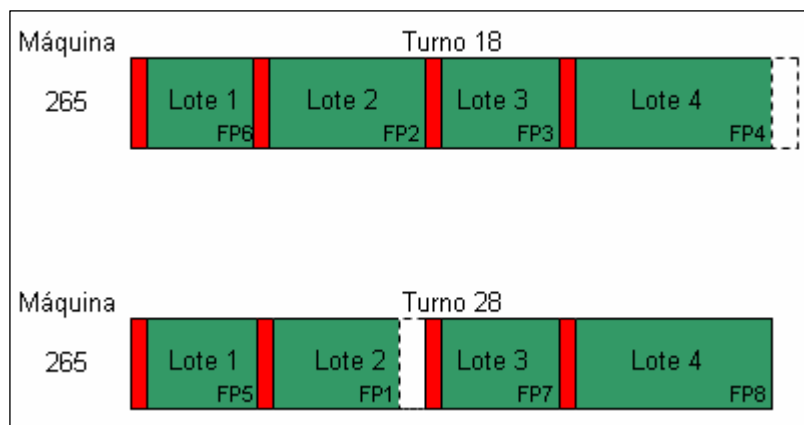


Figura 8.7 - Sequenciamento após a troca de dois lotes.

Nota-se que, a partir da simples inserção dos dois lotes, estourou a capacidade de tempo do turno 18 e existe uma capacidade ociosa no turno 28. Para a etapa final de redefinição dos lotes, os novos lotes são formados de acordo com a capacidade de processamento do turno e das FPs. A Figura 8.8 apresenta os turnos após a redefinição dos lotes de produção e FPs.

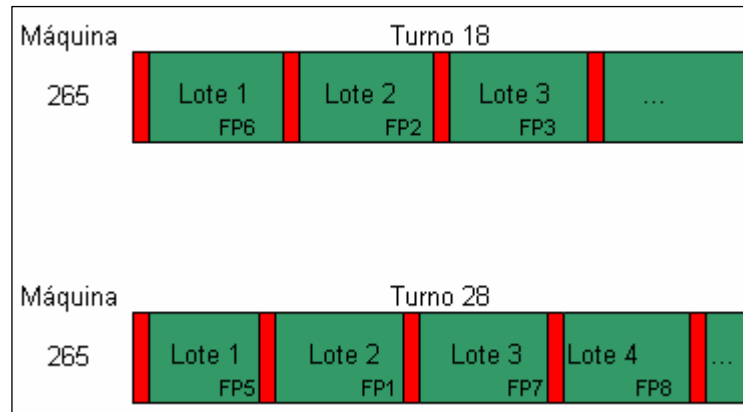


Figura 8.8 - Redefinição dos lotes nos turnos.

Quando na redefinição dos lotes, ocorrerem lotes vizinhos com a mesma FP, gera-se um novo lote com a união dos lotes vizinhos respeitando as datas de entrega e processamento da parte.

Para Gómez [Gómez, 1996], as Famílias de Partes quanto à designação de seus componentes continuam inalteradas. Sobre o ponto de vista do problema do atraso, observa-se que os lotes são utilizados como artifício que permite diminuir o número de conflitos entre a definição de uma seqüência de partes a minimizar o tempo de produção e a influência no tempo de paradas. A Figura 8.9 ilustra o movimento de trocas de lotes.

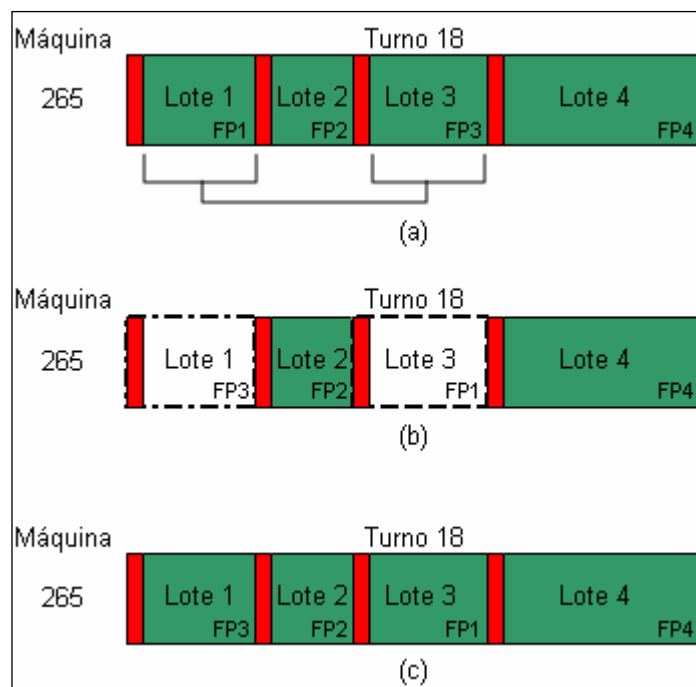


Figura 8.9 - Movimento de troca de lotes no mesmo turno.

Para a Figura 8.9 existem algumas pressuposições que são importantes a serem observadas, sendo que: (a) o número de lotes são os mesmos, (b) os lotes vizinhos pertencem à FPs diferentes, (c) o tempo de processamento do turno é respeitado e (d) os tempos de paradas são os mesmos.

8.1.4.a.3 Movimento de retirada e inserção de parte

O movimento de retirada e inserção de parte consiste na retirada de uma parte qualquer de um lote e a sua inserção num lote de mesma FP. Este tipo de movimento pode refletir na redução do tempo de atraso em virtude da inserção da parte poder ocorrer num lote que o antecede. Segundo Gómez [Gómez, 1996], o movimento de retirada e inserção de partes possui três etapas: (a) retirada da parte referente um lote origem, (b) inserção da parte num lote destino e a (c) redefinição dos lotes.

O processo de inserção da parte pode simplesmente entrar num lote destino em que a FP seja igual. Neste caso, leva-se em consideração somente o tempo de processamento do turno e a precedência das operações da parte à ser processada. A Figura 8.10 ilustra a primeira etapa.

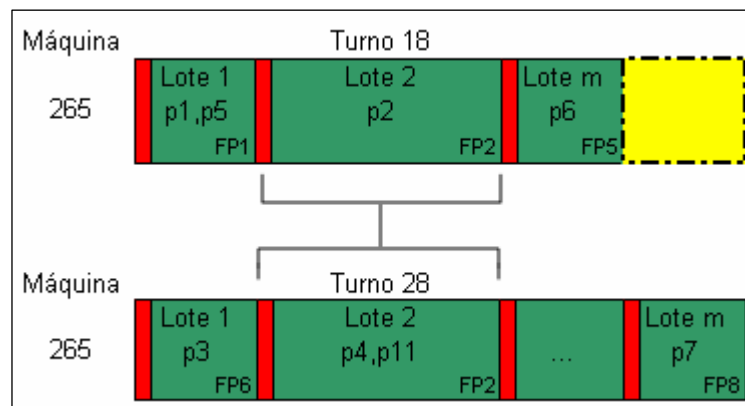


Figura 8.10 - Movimento de retirada da parte no turno 18.

Conforme apresentado na figura acima, o turno 18 possui ociosidade ao final do seu tempo de processamento. O objetivo é a retirada da parte p2 do lote 2 deste turno, e inseri-la no lote 2 do turno 28. Nota-se que ambos os lotes possuem FPs similares. Com a retirada da parte p2, ocorre o aumento da ociosidade do turno 18. A Figura 8.11 apresenta o movimento de inserção da parte no turno 28.

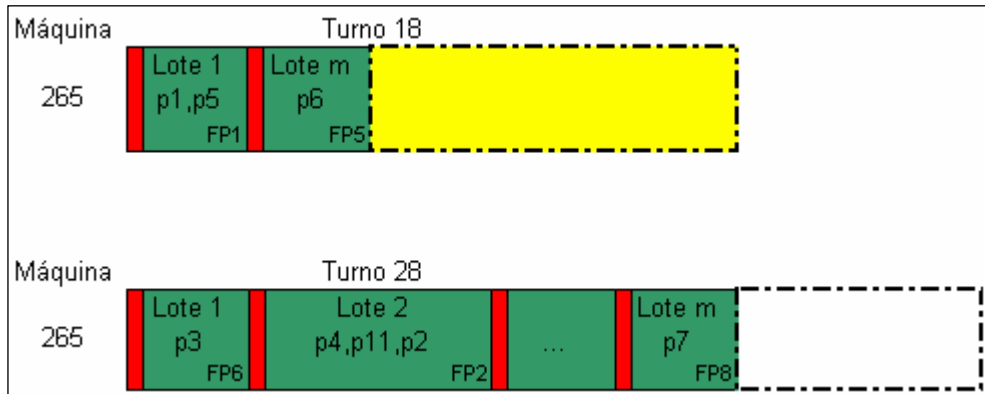


Figura 8.11 - Movimento de inserção da parte no lote destino.

A partir da inserção da parte p2 no turno 28, o tempo de processamento do turno é ultrapassado. É importante realizar o processo de re-escalonamento.

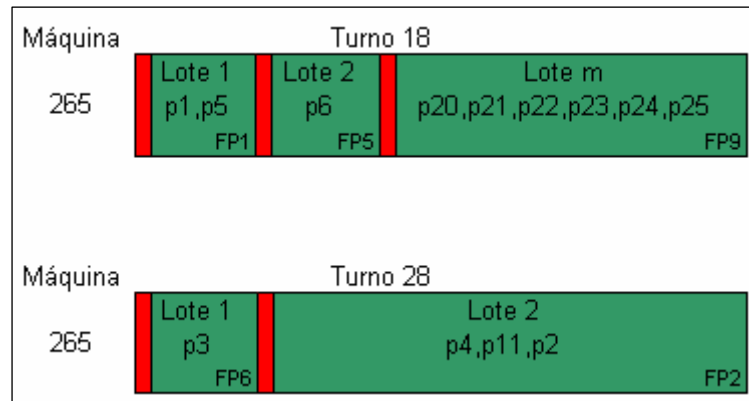


Figura 8.12 - Redefinição de lotes.

Na última etapa, ocorre o re-escalonamento das partes respeitando os tempos de processamento de cada turno e a precedência de operações das partes. Para o turno 18, antes com ociosidade ao final do turno, é ocupado com um lote “m” do turno posterior. Já no turno 28, o lote “m” fora substituído pelo lote 2.

8.1.4.a.4 Lista Tabu

A Lista Tabu é um mecanismo utilizado para evitar ciclos e/ou mínimos locais, ou seja, dado que uma solução encontrou um mínimo local a partir de uma vizinhança, a Lista Tabu evita que, a partir de uma nova vizinhança, encontre esta mesma solução (mínimo local), forçando assim a exploração de vizinhanças até então desconhecidas (estratégia de diversificação) [Glover, 1997]. A Figura 8.13 ilustra o mínimo local.

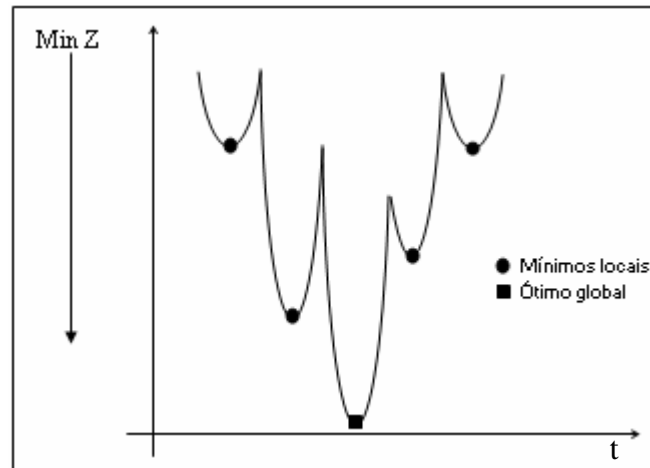


Figura 8.13 - Mínimos locais e ótimo global.

A Lista Tabu é responsável por guardar os movimentos já realizados e está configurada para considerar os últimos 15 movimentos. A primeira lista tabu guarda os movimentos de troca de dois lotes e, a segunda lista tabu, é responsável por guardar o movimento de retirada e inserção da parte. Ambas as listas verificam o movimento reverso [Gómez, 1996]. Se o movimento é tabu, então o movimento não é aceito. A função de critério de aspiração pode perdoar o movimento tabu caso este movimento forneça melhoria na função objetivo. O critério de aspiração utilizado para ambas as listas é o mesmo, ou seja, está baseado na função objetivo.

8.1.4.a.5 Número de iterações

O critério de parada para o algoritmo BT é baseado em duas circunstâncias, sendo que o primeiro momento é o número máximo de iterações (nbmax) sem que ocorra melhora na função objetivo ou até que a função objetivo alcance um valor ótimo conhecido (fmin). Para o modelo proposto é considerado como critério de parada o número máximo de iterações (nbmax). Este parâmetro do BT age como estratégia de intensificação.

8.1.5 Gravação do Plano de Produção

Após a geração do escalonamento final obtida pela quarta fase, a última fase compreende na gravação deste escalonamento no Banco de Dados. O escalonamento gerado é considerado um plano de produção, proporcionando assim obter os históricos e acompanhamentos diários da produção. A Tabela 8.9 mostra o plano de produção.

9 Experimentos e Validação

Neste capítulo são descritos os experimentos e a validação do modelo. A realização dos experimentos se deu através de um estudo de caso no ramo metal-mecânico. Para a etapa de validação, foram considerados trabalhos que propõem a solução do JSSP e um comparativo da qualidade das respostas obtidas do estudo de caso em relação ao trabalho realizado por Gómez [Gómez, 1996].

O modelo proposto foi implementado através da linguagem Delphi 7 em ambiente Windows XP. Para a execução da aplicação foi utilizado um Athlon XP 2600+ com 512mb RAM e o banco de dados Oracle 8i.

9.1 Experimento

Para os experimentos foram considerados quatro estratégias de otimização, sendo elas: (a) o tempo total de produção (*makespan*), (b) o tempo total de atraso, (c) o tempo total de paradas e (d) o tempo total de ociosidade. É importante ressaltar que, os pesos para cada variável de decisão foram variados de forma a comprovar a qualidade dos resultados obtidos e também refletir o conflito entre a flexibilidade e produtividade. Os experimentos realizados seguiram as seguintes etapas:

- contextualização do estudo de caso;
- parâmetros considerados no modelo;
- solução inicial;
- solução real de produção (escalonamento obtido através do estudo de caso);
- solução não-tendenciosa;
- solução tendenciosa que privilegie a variável de decisão *makespan* (peso p_1);
- solução tendenciosa que privilegie a variável de decisão atraso (peso p_2);
- solução tendenciosa que privilegie a variável de decisão parada (peso p_3);
- solução tendenciosa que privilegie a variável de decisão ociosidade (peso p_4); e
- variação dos parâmetros *nbmax* e tamanho da lista tabu.

9.1.1 Estudo de caso

O objetivo deste trabalho é de apresentar um modelo que seja amplamente aplicável em indústrias em que a produção tenha características de um *Job-shop* e de Manufatura Flexível. Para isso, o modelo é aplicado em um estudo de caso traçando um paralelo entre os resultados efetivos de produção com os resultados obtidos através do modelo.

9.1.1.a O ambiente

O ambiente considerado no estudo de caso é do ramo metal-mecânico e possui a seguinte estrutura de produção conforme ilustrada na Figura 9.1. A descrição completa dos ícones utilizados na Figura 9.1 é abordada na seção 7.1.

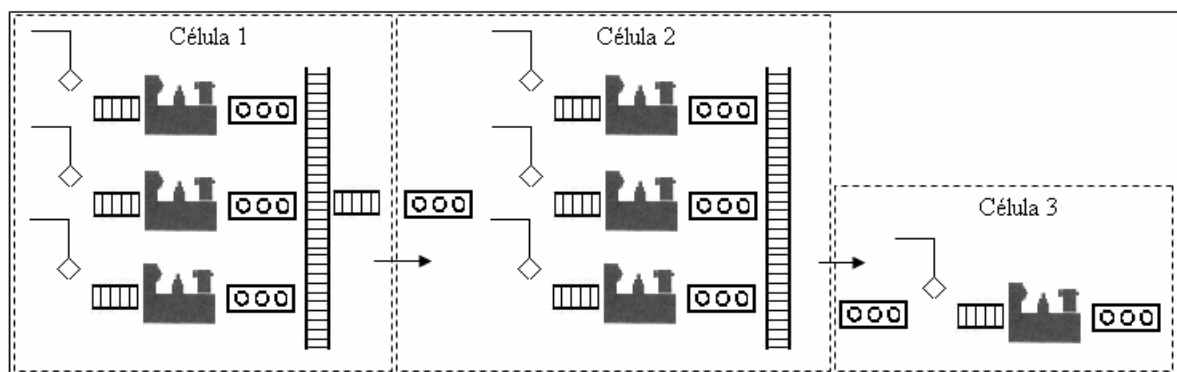


Figura 9.1 - Ambiente de Manufatura para o estudo de caso.

O estudo de caso apresenta três células de manufatura, onde cada célula é responsável por uma operação na rota de produção da parte. O leiaute apresentado está de acordo com as rotas das partes, ou seja, a organização fabril respeita a similaridade dos processos que cada parte deve passar. Os detalhes de cada célula de manufatura são mostrados na seção 7.1.

Todo o controle da produção é realizado através de um sistema computacional que auxilia na programação das partes. Este sistema computacional está interligado através de uma rede *Ethernet* e utiliza um banco de dados para o gerenciamento da produção.

No ambiente de SMF as partes são descritas através de uma série de informações, sendo que estas informações são separadas em duas fichas técnicas:

- ficha de produto: são informações referentes as características da parte, como o peso bruto, peso líquido, dureza padrão, bitola, tipo de material utilizado e padrão de quebra; e
- ficha de processo: são informações referentes as fases, operações, tempo padrão da parte para cada máquina e as máquinas habilitadas para produzir a parte (rota padrão).

A união destas fichas é denominada de “Ficha de Projeto” [Kusiak, 1994; Groover, 2001].

A organização dos turnos de produção se dá em períodos durante o dia, sendo que são definidos três períodos de oito horas, totalizando três turnos de produção. As máquinas são habilitadas para trabalhar de acordo com o dia e turno através do Calendário fabril, que por sua vez considera o dia, turno, horário inicial, horário de almoço, horário final e manutenção. Através do Calendário fabril é possível determinar turnos menores para o final de semana, sendo possível admitir hora-extra na produção e turnos aos finais de semana.

O escalonamento das partes é denominado de plano de produção e consiste da programação de produção. Está dividido em lotes de produção que por sua vez são conjuntos de partes da mesma FP sem que haja necessidade de trocas de ferramentas, material ou outro tipo de parada. A troca dos lotes de produção ocasiona a parada da produção [Kusiak, 1994]. Neste ambiente de SMF a parada de produção é caracterizada pelos *setups* de troca de ferramenta, troca de material e troca de tamanho (ajuste da ferramenta), além da parada para troca de turno. A Tabela 9.1 ilustra os tipos de paradas com os seus tempos respectivos e a Tabela 9.2 mostra a estrutura do calendário fabril.

Tabela 9.1 - Paradas utilizadas no ambiente de manufatura

Tipo de paradas	Tempo médio (minutos)
Troca de material (<i>setup</i>)	15
Troca de ferramenta (<i>setup</i>)	8
Troca de tamanho (<i>setup</i>)	5
Troca de turno	3

É fundamental entender que o *setup* é considerado um tipo de parada e consiste no tempo utilizado na máquina entre dois lotes [Kusiak, 1994]. Na Tabela 9.1 é possível visualizar quatro paradas, sendo que três são *setups*. Alguns autores consideram que a troca de turno também é um *setup*, porém, para nosso modelo foi utilizado esta distinção justamente para contemplar em outros ambientes de manufatura paradas do tipo: falta de material, falta de pedidos e falta de energia elétrica.

Tabela 9.2 - Calendário fabril

Atributo	Valor
Máquina	275
Turno	18
Dia	20/02/2006
Manutenção	Não
Hora inicial	00:00
Hora final	08:00
Hora almoço inicial	05:00
Hora almoço final	06:00

A Tabela 9.2 apresenta a estrutura do calendário fabril com os atributos de alocação. É a partir desta estrutura que o modelo realiza o escalonamento das partes (dimensão temporal do modelo).

9.1.1.b O produto

Todos os produtos possuem a ficha de projeto que por sua vez define as características técnicas e de processo. A partir das informações técnicas contidas nesta ficha, é possível determinar que tipo de parada ocorrerá, como por exemplo: o tamanho da parte, a matéria-prima utilizada e a ferramenta, são informações provenientes da ficha de projeto. Desta forma, tem-se mais vantagem para o gerenciamento do escalonamento visando a redução a parada. A Figura 9.2 mostra as características técnicas de um produto produzido no ambiente contextualizado.

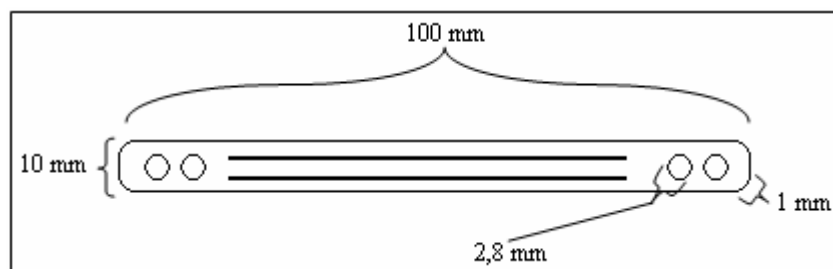


Figura 9.2 - Características técnicas do produto alma de aço.

Através da figura acima, é possível identificar algumas das informações contidas na ficha de projeto, ou seja, as medidas do produto. Cabe lembrar que, a política de geração dos lotes está baseada no agrupamento de partes dado que não ocorram paradas. A partir deste princípio, uma das paradas mencionadas na Tabela 9.1 diz respeito à troca de tamanho e a

outra está relacionada com a troca do material. Por isso, a importância das informações contidas na ficha de projeto para a geração do escalonamento, pois neste caso, é possível identificar o tamanho do material e a espessura da matéria-prima utilizada, e ambos dizem respeito à parada para troca de tamanho e troca de material, respectivamente. A parada para troca de ferramenta está relacionada com o número de furações da parte, neste caso, quatro furações. O exemplo da Figura 9.2 mostra o comprimento de 100mm, a largura de 10mm, a espessura do material de 1mm e a furação de 2,8mm com quatro furações.

9.1.2 Parâmetros considerados no modelo

A seguir são apresentados os parâmetros de configuração para o algoritmo BT e os atributos do ambiente mencionado no estudo de caso.

Para a realização dos experimentos, foram consideradas as seguintes informações:

- o tempo de processamento de um turno é de 480 minutos, totalizando ao final do dia em 1440 minutos;
- cada parte possui uma rota padrão (restrição para o *Job-shop*);
- uma máquina pode somente processar uma parte por vez (restrição para o *Job-shop*);
- uma parte pode somente ser processada em uma máquina por vez (restrição para o *Job-shop*);
- não existe quebra de rota (restrição para o *Job-shop*);
- o processamento de uma parte na máquina não pode ser interrompido, ou seja, processamento não-preemptivo (restrição para o *Job-shop*);
- as rotas são variadas (característica da Manufatura Flexível);
- as partes são diversificadas, ou seja, *mix* de partes (característica da Manufatura Flexível);
- os lotes são diversificados (característica da Manufatura Flexível);
- seleção do plano de produção efetivo (dados reais de produção) contendo 200 partes divididas em 7 máquinas;
- o número máximo de iterações sem sofrer melhora na solução (nbmax) fixado em 100 (algoritmo BT); e
- o tamanho da lista tabu fixado em 15 (algoritmo BT).

9.1.3 Solução inicial

A solução inicial foi obtida através da aplicação das fases de extração dos dados de produção proveniente das informações do estudo de caso, da geração das FPs (Eficácia de ~98%) e da aplicação da regra de despacho. As informações são mostradas na Tabela 9.3.

Tabela 9.3 – Solução inicial

Máquina	<i>Makespan</i>	Atraso	Parada	Ocioso	Instantes de Paradas	Número de Atrasos
226	4783	4320	253	554	27	3
265	388	1440	160	0	17	1
301	706	1440	361	1	39	1
312	5004	4320	459	776	49	3
400	673	1440	329	15	35	1
501	5255	14400	780	572	84	10
515	424	2880	189	0	20	2
Total	5255	30240	2530	1918	271	21

É possível verificar a magnitude (em minutos) do resultado obtido em cada variável de decisão. A variável de decisão atraso possui valores bem maiores do que as demais, dado que a cada 1 dia de atraso, corresponda a 1440 minutos. Nesta solução inicial, observa-se que o atraso está sendo privilegiado, sendo assim, as datas de entrega não estão sendo atendidas, totalizando em 21 partes com atraso (0,11% das partes estão atrasadas). A Tabela 9.4 ilustra o escalonamento real de produção obtido através do histórico de produção. Cabe lembrar que, a solução real de produção não sofreu nenhuma otimização, de forma a manter o escalonamento utilizado pelo estudo de caso.

Tabela 9.4 - Solução obtida através do escalonamento real de produção

Máquina	<i>Makespan</i>	Atraso	Parada	Ocioso	Instantes de Paradas	Número de Atrasos
226	4544	1440	304	527	33	1
265	361	1440	199	0	21	1
301	649	1440	473	1	51	1
312	4704	1440	665	729	71	1
400	626	1440	441	14	47	1
501	4940	2880	952	538	102	2
515	403	2880	240	0	26	2
Total	4940	12960	3273	1808	351	9

Considerando os dados reais de produção, apresentado na Tabela 9.4, é possível verificar que o atraso é bem menor em relação à solução inicial (Tabela 9.3). Contudo, as paradas de produção são bem maiores do que a solução obtida anteriormente (29,5%). É importante entender o tempo menor do atraso e o aumento das paradas de produção em relação à solução inicial, pois neste caso, o escalonamento real está levando em conta a entrega pontual das partes, desconsiderando a otimização da variável de decisão parada.

A partir da Tabela 9.4, também é possível verificar que a regra de despacho utilizada é por data de entrega, em virtude do baixo número de atrasos e da baixa produtividade decorrente do aumento de paradas (produção fragmentada). A Figura 9.3 ilustra a relação entre a solução real e a solução inicial.

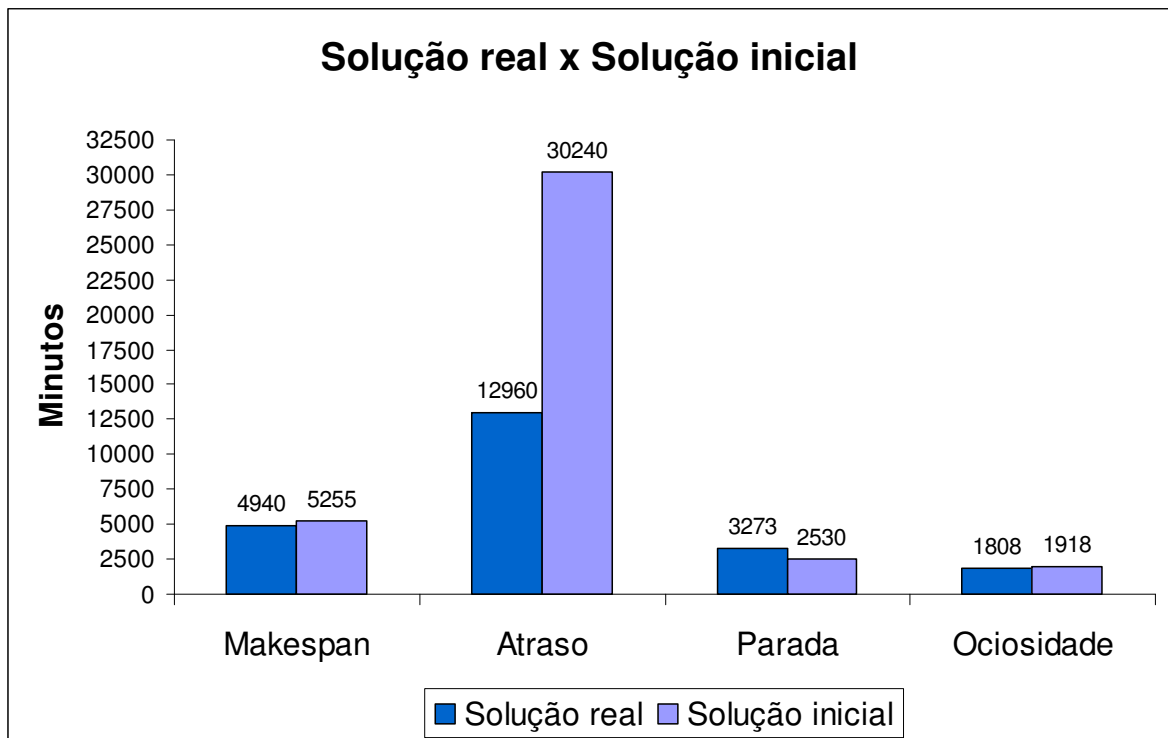


Figura 9.3 - Gráfico solução real x solução inicial.

Através da relação das soluções foi possível verificar a grande diferença da variável atraso. Esta diferença ocorre em virtude das políticas adotadas para a geração de cada solução, onde, a solução inicial utiliza a regra de despacho FPs e DEMR e a solução real utiliza a regra de despacho DEMR. Cabe lembrar que, a variável de decisão atraso possui uma magnitude maior do que as demais devido ao atraso de 1 dia ser considerado 1440 minutos. Nesta etapa dos experimentos, obteve-se uma solução inicial que, embora não seja melhor que a solução real utilizada pela indústria deste estudo de caso, tem-se grande utilidade nas próximas etapas.

9.1.4 Solução tendenciosa e não-tendenciosa

Uma vez que a solução inicial e a solução real para o estudo de caso são conhecidas, é fundamental definir os pesos para cada variável de decisão da função objetivo. Foram consideradas três soluções para os experimentos a partir desta etapa, sendo: (a) a solução tendenciosa, (b) a solução não-tendenciosa (obtida através da solução inicial) e (c) a solução real. A solução não-tendenciosa é aquela em que o peso não privilegie nenhuma das variáveis de decisão da função objetivo. Já a solução tendenciosa é aquela em que os pesos são escolhidos objetivando o benefício de determinada variável de decisão [Mood, 1974].

Para a solução não-tendenciosa, foram gerados aleatoriamente valores para cada peso, seguindo uma distribuição normal com intervalo [0, 100]. Foram realizados 100 experimentos utilizando a fase quatro da arquitetura, que é a otimização do escalonamento através do algoritmo BT. A Tabela 9.5 mostra as médias obtidas após os experimentos.

Tabela 9.5 - Médias obtidas após os experimentos

Máquina	<i>Makespan</i>		Atraso		Parada		Ociosidade	
	Média	σ	Média	σ	Média	σ	Média	σ
226	4227,62	45,75	2966,40	57,60	238,15	12,10	494,29	8,19
265	358,82	3,88	1483,20	28,80	156,96	7,98	0,00	0,00
301	646,49	7,00	0,00	0,00	351,81	17,88	1,03	0,02
312	4347,57	47,05	1483,20	28,80	424,34	21,56	679,91	11,27
400	610,92	6,61	0,00	0,00	318,26	16,17	13,33	0,22
501	4451,25	48,17	2966,40	57,60	703,63	35,75	489,16	8,11
515	381,19	4,13	0,00	0,00	180,78	9,19	0,00	0,00
Total	4451,25	48,17	8899,20	172,80	2373,92	120,62	1677,72	27,81

A partir das médias amostrais obtidas dos 100 experimentos, foi possível gerar uma solução não-tendenciosa utilizando a média da variável de decisão atraso. A partir desta média, encontrou-se a proporção das outras variáveis através da divisão da média de cada variável pela média da variável atraso. Os pesos proporcionais foram os seguintes: $p_1 = 2$, $p_2 = 1$, $p_3 = 3,75$ e $p_4 = 5,30$. A Tabela 9.6 ilustra os resultados obtidos pela fase quatro da arquitetura utilizando os pesos proporcionais, ou seja, a solução não-tendenciosa;

Tabela 9.6 - Solução não-tendenciosa

Máquina	Makespan	Atraso	Parada	Ociosidade	Instantes de Paradas	Número de Atrasos
226	4616	2880	239	546	25	2
265	381	1440	157	0	17	1
301	681	0	351	1	37	0
312	4845	2880	443	751	47	2
400	649	0	315	14	33	0
501	5040	4320	748	554	79	3
515	411	0	184	0	19	0
Total	5040	11520	2436	1866	258	8

A Tabela 9.6 apresenta a solução não-tendenciosa, ou seja, a solução que não privilegia nenhuma variável de decisão. Com a solução não-tendenciosa, ocorreu a redução do conflito entre as variáveis atraso e parada, ou seja, a relação entre as paradas de produção e as datas de entrega. A Figura 9.4 ilustra a relação da solução real e não-tendenciosa.

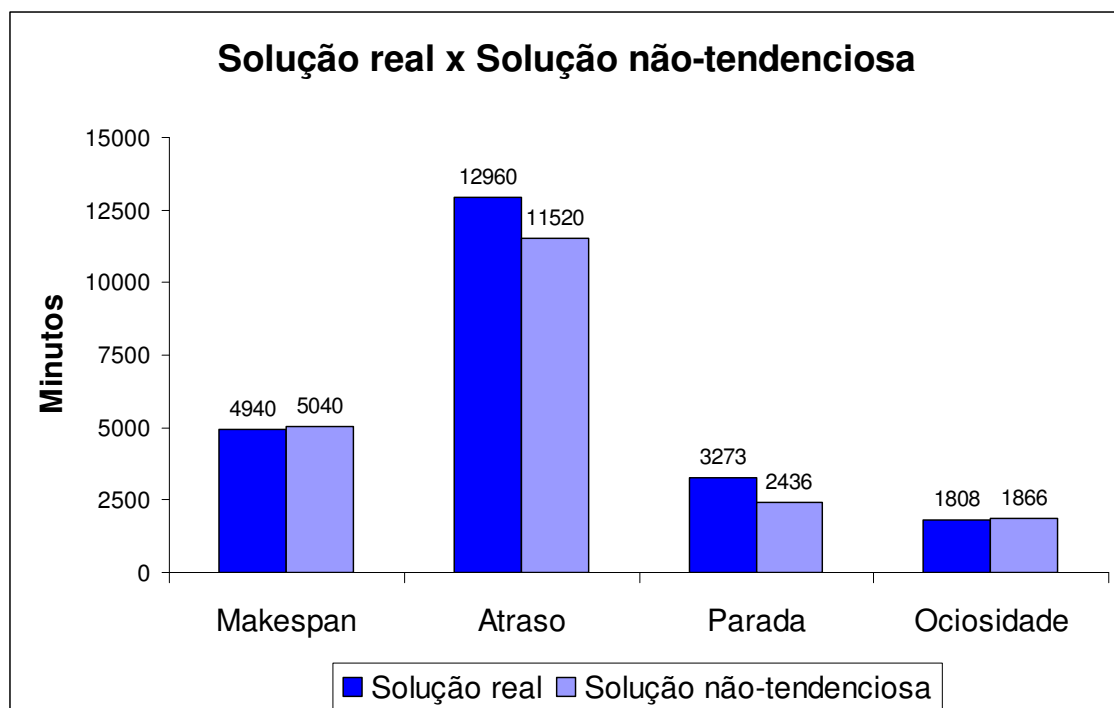


Figura 9.4 - Gráfico solução real x não-tendenciosa.

Apesar das variáveis de decisão não possuírem diferença significativa entre as soluções, através da solução não-tendenciosa foi possível obter o escalonamento ótimo conhecido até esta etapa dos experimentos, superando a solução real. Dessa forma, tem-se

uma solução (não-tendenciosa) que possa ser colocada em prática para outros planos de produção e aplicada como uma estratégia de escalonamento para este estudo de caso.

Para as próximas etapas de experimentos, são realizadas diversas estratégias de otimização através da variação dos pesos de cada variável de decisão.

9.1.5 Otimização do Tempo Total de Produção

Foram realizados 20 experimentos objetivando a minimização do tempo total de produção (*makespan*) através da variação do peso p_1 . Para as demais variáveis, foram utilizados o peso obtido através da solução não-tendenciosa. A Tabela 9.7 ilustra o peso atribuído para cada um dos 20 experimentos e os valores obtidos pelas variáveis de decisão.

Tabela 9.7 - Minimização do *makespan*

Experimento	P_1	<i>Makespan</i>	Atraso	Parada	Ociosidade
1	2	5040	11520	2436	1866
2	3	5040	11520	2436	1866
3	5	5040	11520	2436	1866
4	10	5040	11520	2436	1866
5	15	5040	11520	2436	1866
6	20	4788	12960	2314	1773
7	25	4788	12960	2314	1773
8	30	4788	12960	2314	1773
9	35	4788	12960	2314	1773
10	40	4788	12960	2314	1773
11	45	4788	12960	2314	1773
12	50	4788	12960	2314	1773
13	60	4379	17280	2193	1636
14	70	4379	17280	2193	1636
15	80	4379	17280	2193	1636
16	90	4379	17280	2193	1636
17	100	4379	17280	2193	1636
18	200	4379	17280	2193	1636
19	300	4379	17280	2193	1636
20	500	4379	17280	2193	1636

Através da Tabela 9.7 é possível verificar que a variável de decisão *makespan* tem o seu melhor valor com o peso a partir de 60. Com a redução do *makespan*, ocorreu a minimização do tempo total de paradas e do tempo ocioso, demonstrando a relação de dependência entre estas variáveis de decisão. A Figura 9.5 compara a solução tendenciosa que privilegiou o *makespan* com as demais soluções.

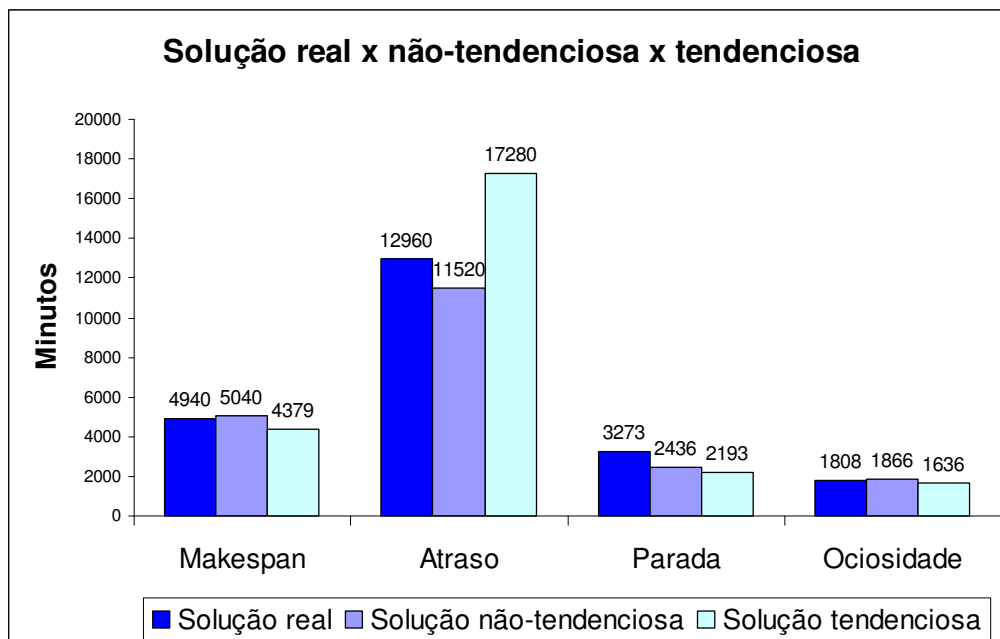


Figura 9.5 - Gráfico solução real x não-tendenciosa x tendenciosa (variação p_1).

O comportamento das soluções é semelhante, dado que a variável *makespan* reduziu o tempo total da produção, conseqüentemente as variáveis ociosidade e parada também diminuíram (dependência). A solução não-tendenciosa obteve um tempo total de produção de 5040 minutos comparado com o tempo da solução tendenciosa de 4379 minutos. A diferença das soluções correspondeu a 13% e, comparado com a solução real, a solução tendenciosa obteve uma diferença de 11%. Ao realizar a análise da variável tempo total de paradas, observou-se um tempo de 2436 minutos da solução não-tendenciosa, comparado ao tempo da solução tendenciosa de 2193, resultando em 10% a menos. Analogamente, para o tempo ocioso dos turnos, obteve 1866 minutos para a solução não-tendenciosa e um tempo de 1636 minutos para solução tendenciosa, tendo uma diferença de 12%. Apesar de obter o valor ótimo conhecido para as variáveis *makespan*, parada e ociosidade, a solução não-tendenciosa continua sendo a estratégia com a melhor solução para todas as variáveis, dado que a solução tendenciosa, mesmo com a magnitude do seu peso ($p_1 \geq 60$) não refletiu o *f* ótimo.

A figura a seguir, mostra as variáveis de decisão e seu comportamento dado o privilégio da variável *makespan*.

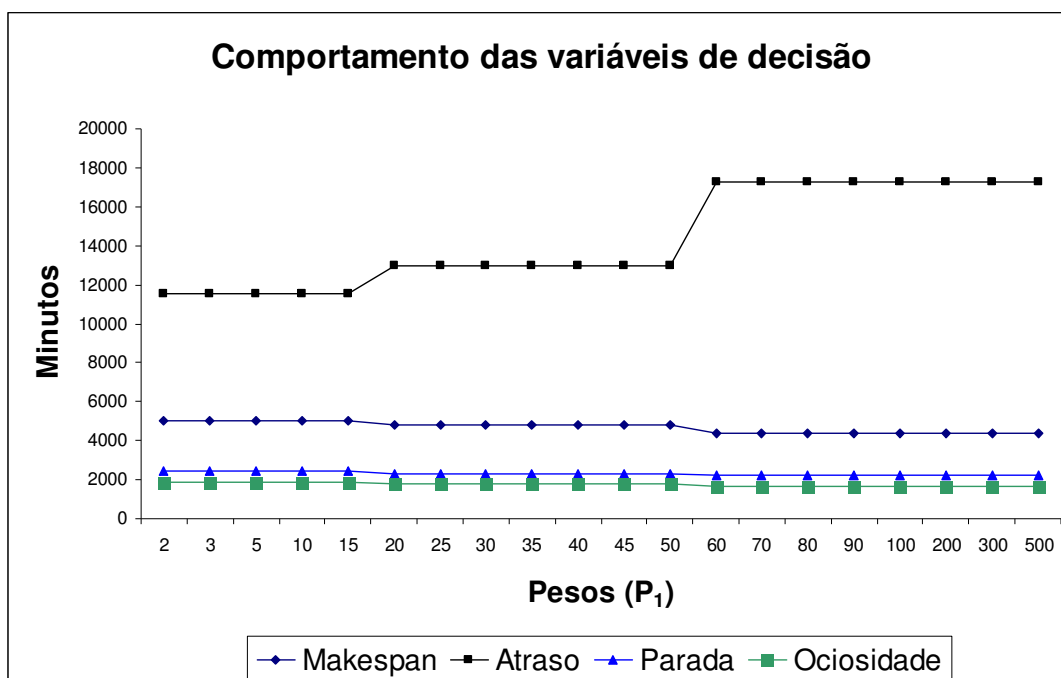


Figura 9.6 - Comportamento das variáveis de decisão dada variação de p_1 .

À medida que o peso de p_1 é aumentado, a variável *makespan* fica mais significativa, enquanto as variáveis parada e ociosidade vão diminuindo e perdendo assim a influência na função objetivo.

Através desta estratégia de minimização do *makespan* foi possível verificar que, à medida que as demais variáveis estão minimizando, a variável do tempo total de atraso vai aumentando, provando que existe o conflito entre respeitar as datas de entrega e a redução das paradas [Gómez, 1996]. Para demonstrar este comportamento, a Tabela 9.8 apresenta 10 experimentos em que o peso de p_2 (atraso) foi desconsiderado.

Tabela 9.8 - Minimização do *makespan* desconsiderando o peso p_2

Experimento	P_1	<i>Makespan</i>	Atraso	Parada	Ociosidade
1	2	5040	31680	2436	1866
2	3	5040	31680	2436	1866
3	5	5040	31680	2436	1866
4	10	4788	38880	2314	1773
5	15	4788	38880	2314	1773
6	20	4788	38880	2314	1773
7	25	4379	41760	2193	1636
8	30	4379	41760	2193	1636
9	40	4379	41760	2193	1636
10	50	4379	41760	2193	1636

Observa-se que as variáveis *makespan*, parada e ociosidade obtiveram os seus melhores resultados com o peso menor do que o peso obtido através da Tabela 9.7 (experimentos que consideraram o peso da variável atraso). Isto ocorre em virtude de não existir relação entre o tempo total de atraso e as demais variáveis, dado que o valor para o seu peso foi desconsiderado. Com o peso $p_2 = 0$, a busca no espaço se dá somente através das variáveis *makespan*, parada e ociosidade.

9.1.6 Otimização do Tempo Total de Atraso

A estratégia de minimização do tempo total de atraso privilegia a entrega das partes, respeitando assim as datas de entrega. Conforme experimentos realizados e objetivando a minimização do tempo total de atraso, foram realizados 20 experimentos variando o peso p_2 de forma a compreender o comportamento desta estratégia de otimização. Para as demais variáveis foram utilizados os pesos da solução não-tendenciosa. A Tabela 9.9 mostra o resultado dos experimentos dada a variação do peso p_2 .

Tabela 9.9 - Minimização do tempo total de atraso

Experimento	P_2	<i>Makespan</i>	Atraso	Parada	Ociosidade
1	1	5040	11520	2436	1866
2	3	5040	11520	2436	1866
3	5	5040	11520	2436	1866
4	10	5040	11520	2436	1866
5	15	5040	11520	2436	1866
6	20	5040	11520	2436	1866
7	25	5040	11520	2436	1866
8	30	5040	11520	2436	1866
9	35	5040	11520	2436	1866
10	40	5040	11520	2436	1866
11	45	5040	11520	2436	1866
12	50	5040	11520	2436	1866
13	60	6199	7200	2996	2053
14	70	6199	7200	2996	2053
15	80	6199	7200	2996	2053
16	90	6199	7200	2996	2053
17	100	6695	4320	3236	2155
18	200	6695	4320	3236	2155
19	300	6695	4320	3236	2155
20	500	6695	4320	3236	2155

Tendo como objetivo a redução dos tempos totais de atraso, obteve-se a solução de 4320 minutos e, até esta etapa dos experimentos, é o menor valor para a variável de decisão atraso, diferenciando-se em 266% em relação à solução não-tendenciosa e 300% em relação à solução real. O fato interessante da variação deste peso está relacionado com o aumento

significativo das demais variáveis, provando que existe o conflito entre atender o cliente, ou seja, respeitar as datas de entrega e a produtividade do sistema de produção.

A melhor solução conhecida para o tempo total de atraso foi obtida através do peso p_2 a partir de 100. Este comportamento é decorrente do aumento significativo da magnitude da variável, e conseqüência das demais variáveis serem menos privilegiadas e perderem influência na função objetivo. Com exceção da variável ociosidade, as demais variáveis sofreram um aumento médio de ~33% em relação à solução não-tendenciosa. A Figura 9.7 mostra o comparativo da solução real, não-tendenciosa e a solução tendenciosa, dada a variação do peso p_2 , e a Figura 9.8 ilustra o comportamento de cada variável de decisão.

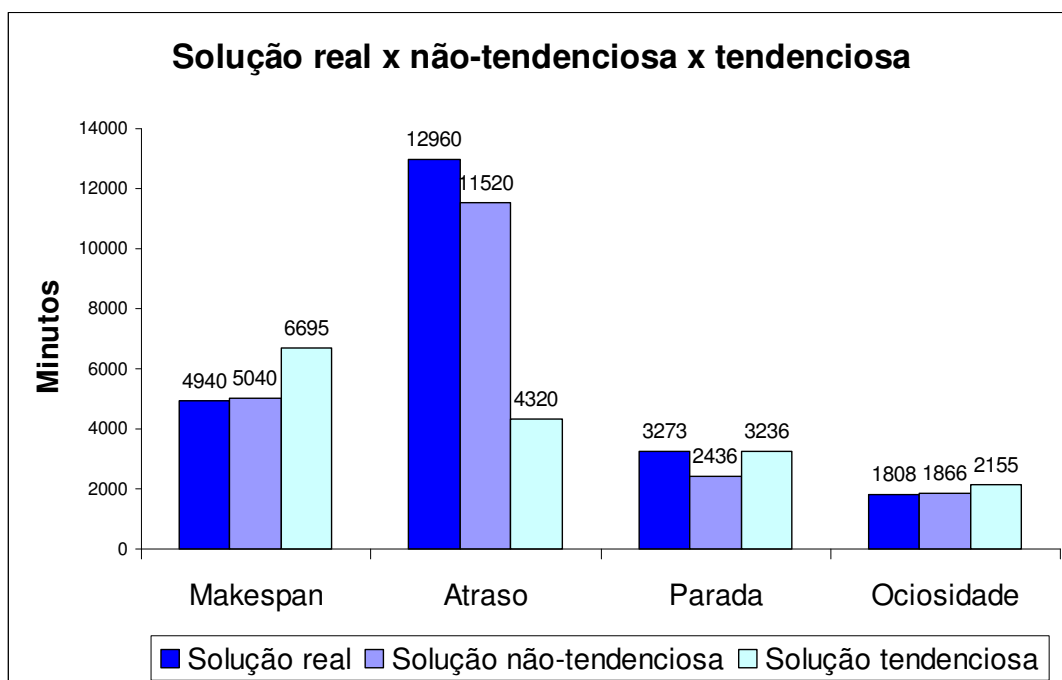


Figura 9.7 - Gráfico solução real x não-tendenciosa x tendenciosa (variação p_2).

O privilégio em relação às datas de entrega resultou em uma diferença significativa no atendimento pontual ao cliente. Porém, o aumento das demais variáveis reflete a queda da produtividade e da eficiência [Slack, 1997].

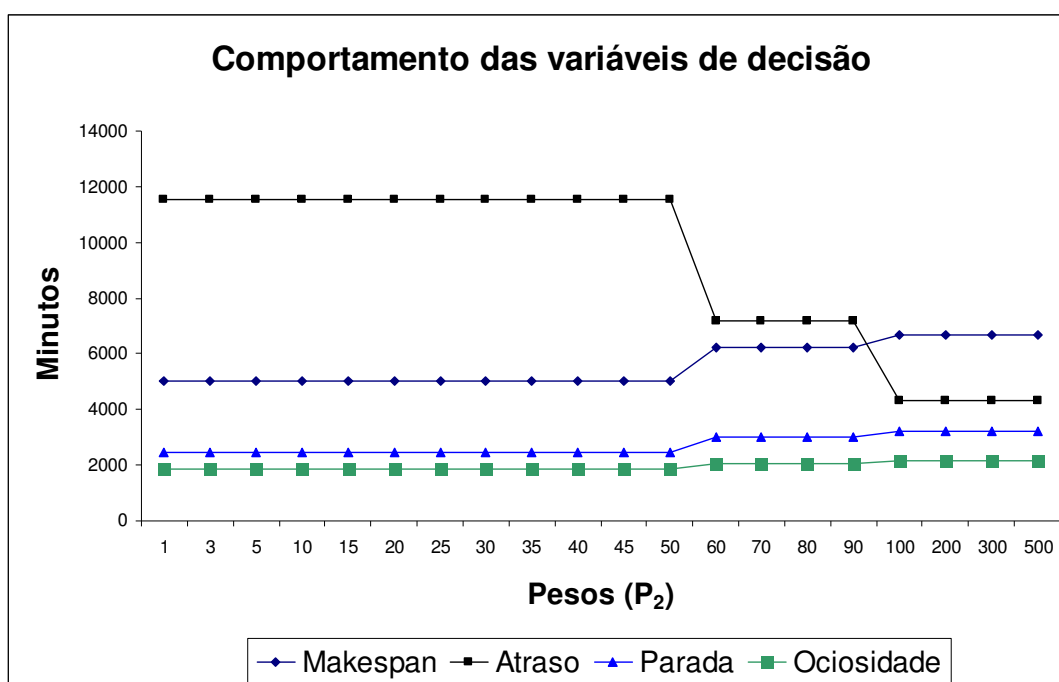


Figura 9.8 - Comportamento das variáveis de decisão dada variação de p_2 .

Constatou-se que a variável atraso obteve redução significativa, superando pela primeira vez a estratégia do tempo total de produção, embora estas variáveis possuam magnitudes diferentes para serem comparadas. Considerando esta estratégia, a melhor solução para a variável de decisão atraso ocorreu com o peso a partir de 100. Outro fator de reflexão está na relação com a variável *makespan*, dado que, à medida que o atraso é reduzido, o *makespan* aumenta (peso a partir de 50).

9.1.7 Otimização do Tempo Total Parado

A produtividade e o *makespan* estão relacionados diretamente com o tempo total de paradas. Este comportamento ocorre em virtude de:

- quanto maior o tempo total de paradas, maior o tempo total de produção;
- quanto maior o tempo total de paradas, menor é a produtividade. A produtividade é medida através do coeficiente entre o tempo de trabalho (tempo disponível – tempo parado) e o tempo previsto para produzir a parte [Slack, 1997]; e
- quanto maior o tempo total de paradas, maior será a fragmentação do escalonamento de produção.

De maneira a entender o comportamento desta variável, foram realizados 20 experimentos em que o peso (p_3) foi variado. Para as demais variáveis foram utilizados os

pesos obtidos através da solução não-tendenciosa ($p_1 = 2$, $p_2 = 1$ e $p_4 = 5,30$). Os resultados são mostrados abaixo.

Tabela 9.10 - Minimização do tempo total parado

Experimento	P_3	<i>Makespan</i>	Atraso	Parada	Ociosidade
1	3,75	5040	11520	2436	1866
2	5	5040	11520	2436	1866
3	8	5040	11520	2436	1866
4	10	4680	14400	2235	1710
5	15	4680	14400	2235	1710
6	20	4680	14400	2235	1710
7	25	4680	14400	2235	1710
8	30	4680	14400	2235	1710
9	35	4680	14400	2235	1710
10	40	4680	14400	2235	1710
11	45	4680	14400	2235	1710
12	50	4680	14400	2235	1710
13	60	4680	14400	2235	1710
14	70	4680	14400	2235	1710
15	80	4680	14400	2235	1710
16	90	4680	14400	2235	1710
17	100	4680	14400	2235	1710
18	200	4680	14400	2235	1710
19	300	4680	14400	2235	1710
20	500	4680	14400	2235	1710

Verificou-se que a melhor solução para este experimento ocorreu com o peso a partir de 10. As demais variáveis estabilizaram em consequência de pouca influência na função objetivo, dado seus pesos. Dessa forma, a contribuição destas variáveis foi pouquíssima e não influenciou para a escolha do movimento do algoritmo BT.

Através da relação já demonstrada entre as variáveis *makespan*, parada e ociosidade, constatou-se que estas variáveis sofreram otimização. O fato interessante desta estratégia é que, a variável que privilegia o atraso aumentou, demonstrando assim o conflito entre as datas de entrega e a produtividade. A solução tendenciosa, dada a variação do peso p_3 , obteve uma redução de 8,99% em relação à solução não-tendenciosa. Já o *makespan* obteve 7,14% de redução, o atraso aumentou em 25% e a ociosidade diminuiu em 8,36%. A Figura 9.9 apresenta o comparativo a partir da variação de p_3 .

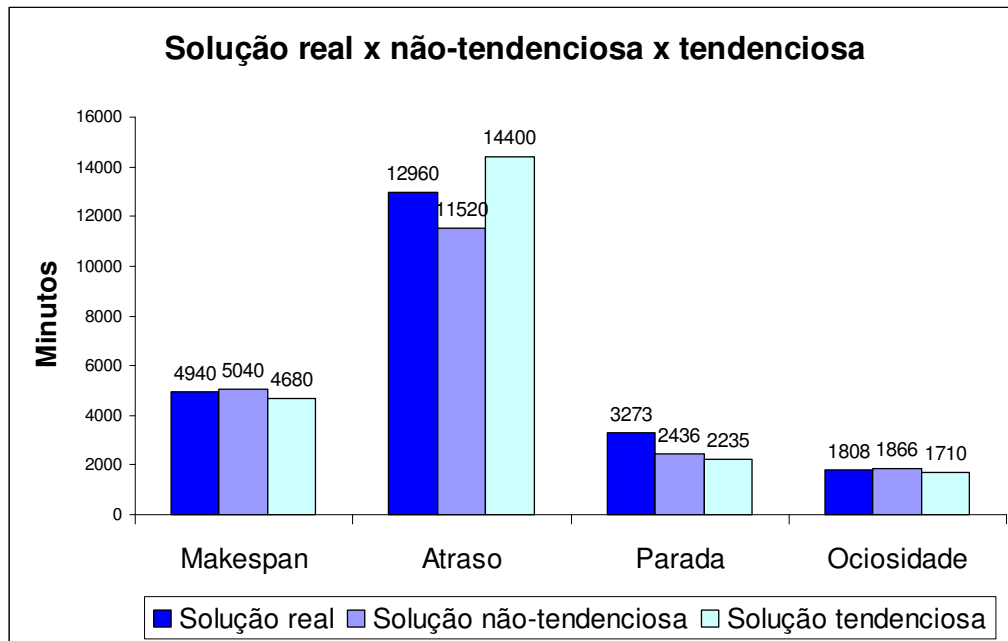


Figura 9.9 - Gráfico solução real x não-tendenciosa x tendenciosa (variação p_3).

A solução tendenciosa que privilegia o tempo total de paradas, foi a solução que obteve os pontos de convergência mais próximos comparados com a solução não-tendenciosa e de rápida otimização ($p_3 \geq 10$), como pode ser observado na figura acima. É importante entender que até o momento, a solução não-tendenciosa havia obtido a melhor estratégia de otimização. Após a execução dos experimentos observou-se que a solução tendenciosa dada variação de p_3 obteve o melhor valor de otimização para este estudo de caso. A Figura 9.10 mostra o comportamento das variações de decisão.

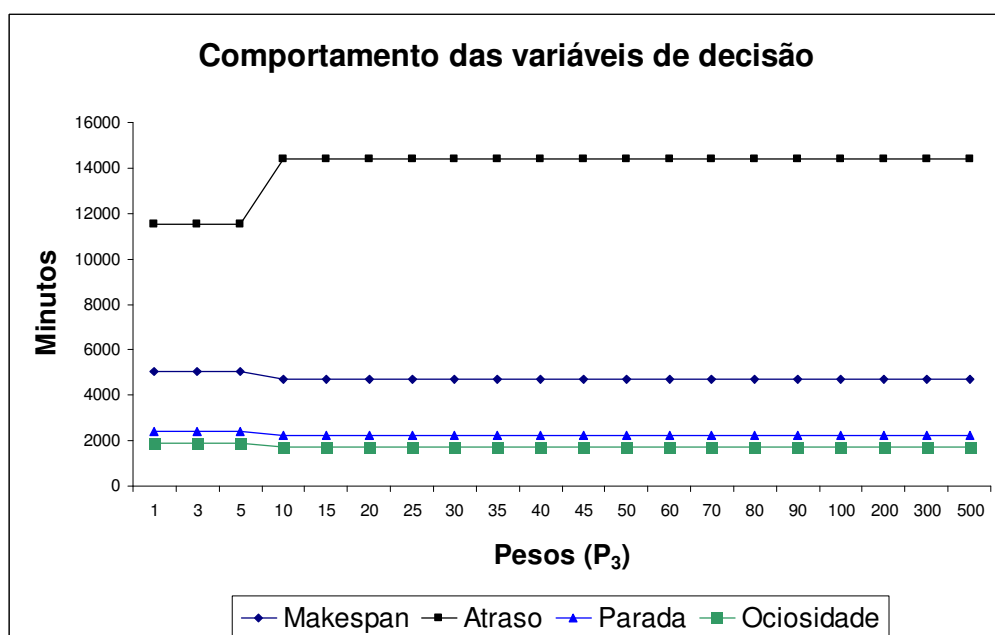


Figura 9.10 - Comportamento das variáveis de decisão dada variação de p_3 .

9.1.8 Otimização do Tempo Total Ocioso

A estratégia de otimização do tempo total ocioso é amplamente utilizada em paralelo com outras variáveis de decisão, tais como: *makespan* e parada. É nesta estratégia que é otimizado o tempo ocioso do turno (a fragmentação do escalonamento devido à ociosidade ao final do turno ou à espera da máquina pelo processamento da máquina anterior).

Alguns autores como Slack em [Slack, 1997] consideram esta estratégia de otimização como um tipo de parada de produção, como por exemplo: parada ao final do turno ou parada por falta de produção. Em alguns ambientes de Manufatura Flexível é de extrema importância, em virtude das características da parte, como por exemplo: existem partes que, ao iniciar o processo de produção, devem ser obrigatoriamente finalizadas no turno em que iniciou, pois há um grande risco desta parte ficar danificada.

Outro exemplo está na necessidade de maior dureza da parte. Para isso, são submetidos a processos de aumento de dureza (processo de têmpera). Portanto, a parte não pode ficar em modo de espera, pois esta pode ficar danificada (ultrapassar a dureza).

De maneira análoga as demais estratégias de otimização, foram realizados 20 experimentos variando o valor do peso p_4 , neste caso, o peso para a variável de decisão ociosidade. Para as demais variáveis foram utilizados os pesos da solução não-tendenciosa. A Tabela 9.11 mostra os resultados obtidos.

Tabela 9.11 - Minimização do tempo total ocioso

Experimento	P_4	<i>Makespan</i>	Atraso	Parada	Ocioso
1	5,3	5040	11520	2436	1866
2	7	5040	11520	2436	1866
3	9	5040	11520	2436	1866
4	10	5040	11520	2436	1866
5	15	5040	11520	2436	1866
6	20	5040	11520	2436	1866
7	25	5040	11520	2436	1866
8	30	5040	11520	2436	1866
9	35	5040	11520	2436	1866
10	40	4895	11520	2448	1710
11	45	4895	11520	2448	1710
12	50	4895	11520	2448	1710
13	60	4895	11520	2448	1710
14	70	4895	11520	2448	1710
15	80	4895	11520	2448	1710
16	90	4895	11520	2448	1710
17	100	4895	11520	2448	1710
18	200	4895	11520	2448	1710
19	300	4895	11520	2448	1710
20	500	4895	11520	2448	1710

A variável ociosidade obteve a melhor solução com peso p_4 a partir de 10. O comportamento desta variável é interessante, pois reduziu o tempo total de produção, porém

aumentou o tempo de parada. Isto reflete que, ao reduzir o tempo ocioso dos turnos, conseqüentemente, o *makespan* é reduzido devido à diminuição dos fragmentos de ociosidade decorrentes da espera da parte ou do tempo não utilizado ao final do turno. Em virtude da variável parada perder influência na função objetivo, seu valor acabou aumentando, mostrando assim que a variável ociosidade não possui relação direta com a variável parada. Esta estratégia de otimização não atingiu a variável atraso, dado que seu valor ficou inalterado. A Figura 9.11 mostra o comparativo das soluções e a Figura 9.12 apresenta o comportamento das variáveis de decisão.

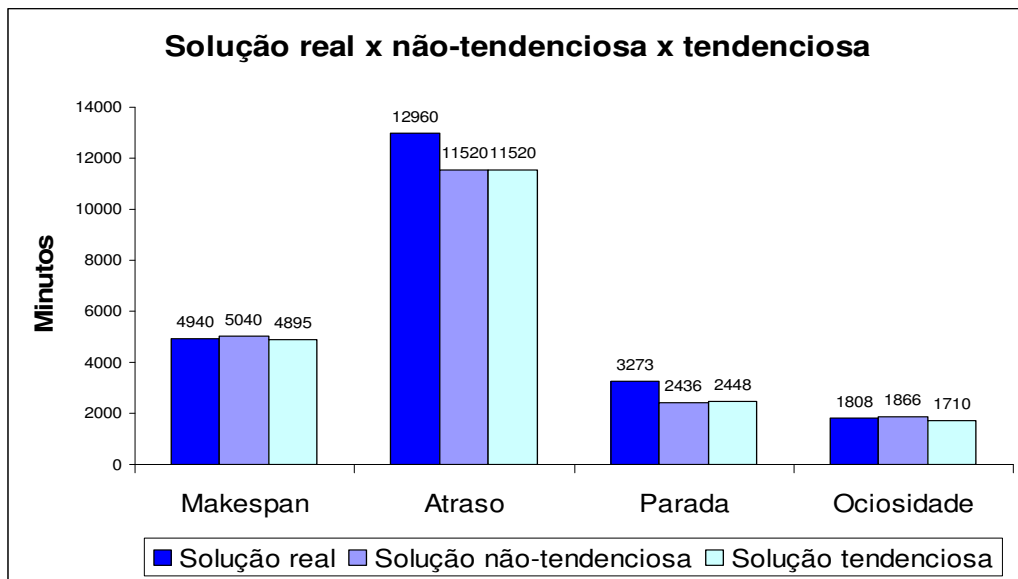


Figura 9.11 - Gráfico solução real x não-tendenciosa x tendenciosa (variação p_4).

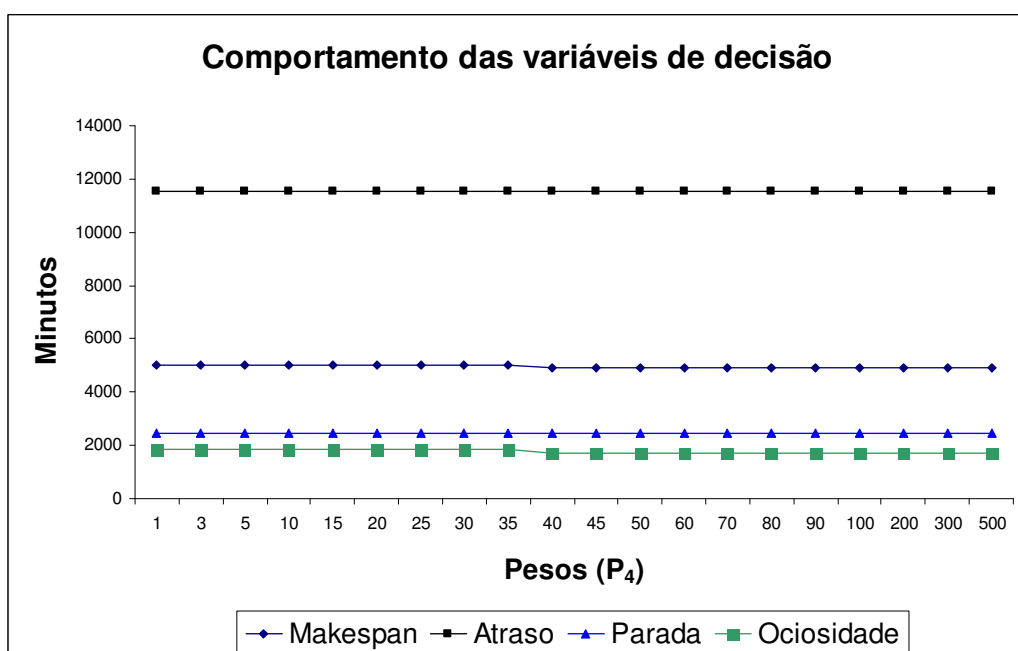


Figura 9.12 - Comportamento das variáveis de decisão dada variação de p_4 .

Através do comportamento ilustrado na Figura 9.12, observa-se que a variável ociosidade compromete diretamente a variável parada. Este comportamento é justificado em virtude da variável parada ter pouca influência na escolha de movimentos do algoritmo BT. Outro reflexo da redução da ociosidade está relacionado com o tamanho dos lotes de produção, visto que, quanto maior o tamanho do lote, maior o tempo ocioso. Apesar do *makespan* ter reduzido para este estudo de caso, não significa que este tenha relação com a estratégia. Caso o tempo de parada seja maior que o tempo ocioso, conseqüentemente, o *makespan* irá aumentar. Neste exemplo, o tempo de parada foi menor que o tempo de ociosidade, refletindo assim na redução do *makespan*.

9.1.9 Lista Tabu e o nbmax

Seguindo as diversas estratégias de otimização, foram realizados 5 experimentos de forma a verificar a influência dos parâmetros do algoritmo BT na função objetivo. As estratégias de otimização foram definidas utilizando os seguintes pesos:

- estratégia A: utilização da solução não-tendenciosa ($p_1 = 2$, $p_2 = 1$, $p_3 = 3,75$ e $p_4 = 5,30$);
- estratégia B: utilização da solução tendenciosa que privilegie o *makespan* ($p_1 = 500$, $p_2 = 1$, $p_3 = 3,75$ e $p_4 = 5,30$);
- estratégia C: utilização da solução tendenciosa que privilegie o atraso ($p_1 = 2$, $p_2 = 500$, $p_3 = 3,75$ e $p_4 = 5,30$);
- estratégia D: utilização da solução tendenciosa que privilegie a parada ($p_1 = 2$, $p_2 = 1$, $p_3 = 500$ e $p_4 = 5,30$); e
- estratégia E: utilização da solução tendenciosa que privilegie a ociosidade ($p_1 = 2$, $p_2 = 1$, $p_3 = 3,75$ e $p_4 = 500$).

Considerando que todos os experimentos anteriores utilizaram o nbmax de 100 e a lista tabu com 15 posições, para o critério de análise, foi configurado o nbmax para iniciar em 100 e a lista tabu em 30 posições. A Tabela 9.12 mostra os valores utilizados para cada parâmetro.

Tabela 9.12 - Valores utilizados para o nbmax e a lista tabu

nbmax	lista tabu ¹	lista tabu ²
100	30	30
200	50	50
300	70	70
500	100	100
700	150	150
900	200	200
1000	300	300
1500	500	500
2000	700	700
3000	1000	1000

As tabelas a seguir apresentam o comportamento de cada estratégia em relação aos valores apresentados na Tabela 9.12.

Tabela 9.13 - Experimentos utilizando a estratégia A

nbmax	lista tabu ¹	lista tabu ²	<i>Makespan</i>	Atraso	Parada	Ociosidade
100	30	30	5040	11520	2436	1866
200	50	50	5040	11520	2436	1866
500	100	100	5040	11520	2436	1866
700	150	150	5040	11520	2436	1866
900	200	200	5040	11520	2436	1866
1000	300	300	5040	11520	2436	1866
2000	700	700	5040	11520	2436	1866

Tabela 9.14 - Experimentos utilizando a estratégia B

nbmax	lista tabu ¹	lista tabu ²	<i>Makespan</i>	Atraso	Parada	Ociosidade
100	30	30	4379	17280	2193	1636
200	50	50	4379	17280	2193	1636
500	100	100	4379	17280	2193	1636
700	150	150	4379	17280	2193	1636
900	200	200	4379	17280	2193	1636
1000	300	300	4379	17280	2193	1636
2000	700	700	4379	17280	2193	1636

Tabela 9.15 - Experimentos utilizando a estratégia C

nbmax	lista tabu ¹	lista tabu ²	<i>Makespan</i>	Atraso	Parada	Ociosidade
100	30	30	6695	4320	3236	2155
200	50	50	6695	4320	3236	2155
500	100	100	4379	17280	2193	1636
700	150	150	7540	1440	4817	3736
900	200	200	4379	17280	2193	1636
1000	300	300	7540	1440	4817	3736
2000	700	700	6695	4320	3236	2155

Tabela 9.16 - Experimentos utilizando a estratégia D

nbmax	lista tabu ¹	lista tabu ²	<i>Makespan</i>	Atraso	Parada	Ociosidade
100	30	30	4680	14400	2235	1710
200	50	50	4680	14400	2235	1710
500	100	100	4680	14400	2235	1710
700	150	150	4680	14400	2235	1710
900	200	200	4680	14400	2235	1710
1000	300	300	4680	14400	2235	1710
2000	700	700	4680	14400	2235	1710

Tabela 9.17 - Experimentos utilizando a estratégia E

nbmax	lista tabu ¹	lista tabu ²	<i>Makespan</i>	Atraso	Parada	Ociosidade
100	30	30	4895	11520	2448	1710
200	50	50	4895	11520	2448	1710
500	100	100	4895	11520	2448	1710
700	150	150	4895	11520	2448	1710
900	200	200	4895	11520	2448	1710
1000	300	300	4895	11520	2448	1710
2000	700	700	4895	11520	2448	1710

A partir dos resultados obtidos para cada estratégia de otimização, constatou-se que o tamanho da lista tabu e o nbmax influenciaram somente a estratégia que busca privilegiar o atraso. Pode-se concluir que a lista tabu é fator importante, pois permite aumentar as estratégias de diversificação da vizinhança de forma a visitar um número maior de soluções. Outro fator importante é a variação do nbmax, pois este parâmetro age como política de

intensificação para a Busca Tabu. A Figura 9.13 ilustra o comportamento da variável de decisão atraso a partir da variação da lista tabu e o nbmax.

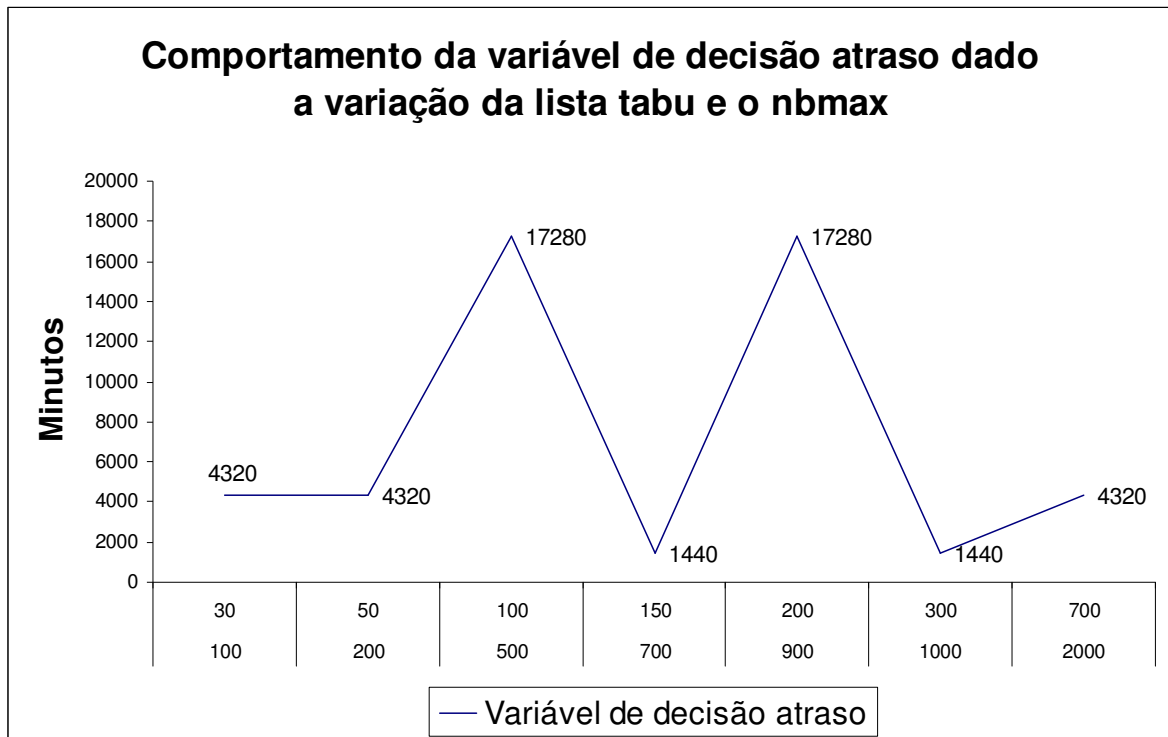


Figura 9.13 – Comportamento da variável atraso dado variação da lista tabu e o nbmax.

9.2 Validação

A etapa de validação do modelo foi realizada através da comparação com trabalhos clássicos que propõem a solução do JSSP [Blazewicz, 1996; Jain, 1998; Zoghby, 2004]. Para testar o desempenho da arquitetura são utilizadas diversas instâncias do JSSP de modo a realizar a validação em diversos ambientes (número de partes x número de máquinas).

Para a validação foram utilizadas as instâncias propostas por Muth [Muth, 1963] denominada de FT06, FT10 e FT20, e as instâncias propostas por Lawrence [Lawrence, 1984] denominadas de LA01, LA06, LA11, LA16, L21, LA29 e LA40. A Figura 9.14 ilustra a estrutura genérica de uma instância.

n	m				
M_{11}	T_{11}	M_{12}	T_{12}	...	M_{1m} T_{1m}
M_{21}	T_{21}	M_{22}	T_{22}	...	M_{2m} T_{2m}
...
M_{n1}	T_{n1}	M_{n2}	T_{n2}	...	M_{nm} T_{nm}

Figura 9.14 - Estrutura genérica de uma instância do JSSP.

A instância genérica para o problema do JSSP possui as seguintes definições: (a) a primeira linha define a dimensão da matriz ($n \times m$), onde o “n” representa o número de partes para o problema em questão e o “m” representa o número de máquinas, (b) as demais linhas representam a rota para cada parte, onde: a variável “M” ilustra a máquina e a variável “T” ilustra o tempo de processamento, (c) as colunas ímpares representam o número da máquina que a parte deverá ser processada e (d) as colunas pares representam o tempo de processamento da parte em cada máquina.

9.2.1 Trabalhos relacionados

Conforme apresentado na seção 5.5, trabalhos relacionados do JSSP, foram selecionados alguns autores em função do tipo de algoritmo utilizado e da sua importância na evolução do estado da arte. Cabe lembrar que, em função da validação do modelo, levou-se em consideração o fato do autor ter resolvido as instâncias selecionadas. A Tabela 9.18 mostra os autores selecionados e o algoritmo utilizado.

Tabela 9.18 - Autores selecionados para a validação do modelo

Autor	Algoritmo	Período
Storer	Busca no espaço	1992
Aarts	Algoritmos genéticos	1994
Dorndorf	Algoritmos genéticos	1995
Nowicki	Busca tabu	1996
Wang	Genético híbrido e têmpera simulada	2001
Aiex	GRASP	2003
Gonçalves	Algoritmos genéticos híbridos	2005

9.2.2 Configuração da validação

Para a configuração do algoritmo BT foi atribuído o valor de 100 para o número máximo de iterações sem obter melhora na função objetivo (nbmax) e para o tamanho da lista tabu, foi atribuído 15.

Em virtude dos autores selecionados utilizarem a função *makespan* para os seus experimentos, os pesos (p_1 , p_2 , p_3 e p_4) para a função objetivo foram fixados em 100, 1, 1 e 1, respectivamente, de modo a privilegiar a variável de decisão *makespan*.

É importante ressaltar que o modelo proposto possui uma arquitetura que tem como objetivo o gerenciamento entre as datas de entrega e o tempo de parada. Por isso, a fase da

geração de FP foi adaptada para agrupar as partes em função das rotas de produção, tal que as partes que possuem as operações similares são agrupadas.

Como as instâncias não possuem as datas de entrega para a parte, e o modelo proposto às utiliza como estratégia de otimização, foram fixadas todas as partes com a mesma data de entrega.

9.2.3 Resultados computacionais

Os resultados da validação são mostrados na Tabela 9.19. Nela é apresentada a instância do problema, a dimensão do problema (número de partes x número de máquinas), a solução ótima conhecida (SOC), a solução obtida pelo modelo proposto (MP) e a solução dos autores selecionados.

Tabela 9.19 - Resultados dos experimentos

Instância	Dimensão	SOC	MP	Gonçalves 2005	Aiex 2003	Wang 2001	Nowicki 1996	Domdorf 1995	Aarts 1994	Storer 1992
FT06	6x6	55	55	55	55	55	55			
FT10	10x10	930	930	930	930	930	930	960	935	952
FT20	20x5	1165	1165	1165	1165	1165	1165	1249	1165	
LA01	10x5	666	666	666	666	666	666	666	666	666
LA06	15x5	926	926	926	926	926	926	926	926	
LA11	20x5	1222	1222	1222	1222	1222	1222	1222	1222	
LA16	10x10	945	945	945	945	945	945	1008	977	981
LA21	15x10	1046	1047	1046	1057	1058	1047	1139	1084	
LA29	20x10	1157	1160	1196	1203		1160	1336	1290	
LA40	15x15	1222	1229	1241	1244		1229	1321	1273	1258

Através da Tabela 9.19 é possível fazer um comparativo entre o modelo proposto e a solução ótima conhecida. Nota-se que o *makespan* com a maior diferença em relação à SOC diferencia-se em 7 unidades de tempo (LA40), equivalendo a uma diferença 0,57%. Constatase que o MP obteve os mesmos resultados obtidos por Nowicki [Nowicki, 1996], que por sua vez utilizou o algoritmo Busca Tabu nos seus experimentos.

De maneira a validar o modelo, os resultados esperados atingiram o objetivo, ou seja, em muitos casos alcançando a SOC e, em outros, aproximando-se. Dessa forma, além de obter os valores ótimos conhecidos, o modelo proposto tem como principal contribuição o gerenciamento entre as datas de entrega e a produtividade do ambiente de produção, visto que as demais abordagens tratam somente o *makespan*. A Figura 9.15 mostra o escalonamento após a aplicação do modelo para a instância FT06.

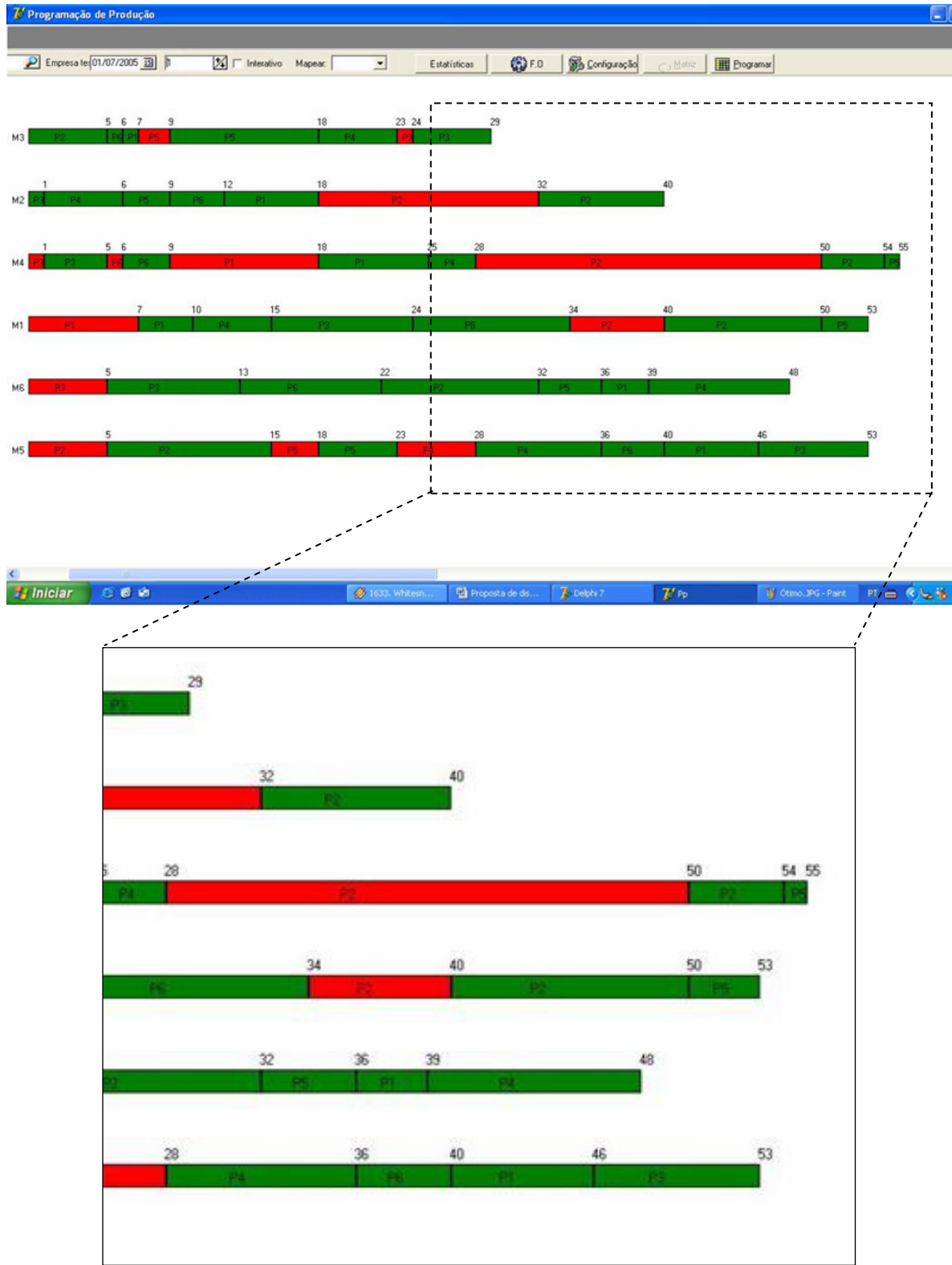


Figura 9.15 - Gráfico Gantt para a instância FT06.

9.2.4 Comparativo com o Modelo de Seqüenciamento de Partes e de Ferramentas

Apesar da dificuldade em encontrar trabalhos que abordam os mesmos aspectos do tipo: *Job-Shop*, SMF, gerenciamento das datas de entrega e as paradas de produção, foi possível comparar o modelo proposto nesta dissertação com o Modelo de Seqüenciamento de Partes e de Ferramentas (MSPF) desenvolvido por Gómez [Gómez, 1996]. O objetivo é analisar o comportamento quanto à qualidade das respostas, dado que Gómez utilizou uma máquina versátil e tinha como objetivo a redução dos instantes de troca e *setup* de modo a respeitar as datas de entrega. A Tabela 9.20 ilustra este comparativo.

Tabela 9.20 – Comparativo modelo proposto e o MSPF

	<i>Makespan</i>		Atraso		Parada		Ociosidade	
	MP	MSPF	MP	MSPF	MP	MSPF	MP	MSPF
Privilegiando o <i>makespan</i>			↑		↓		↓	
Privilegiando o atraso	↑	↑			↑	↑	↑	↓
Privilegiando a parada	↓	↓	↑	↑			↓	↑
Privilegiando a ociosidade	↓	↑	sem variação	↑	↑	↑		

Ao se privilegiar a variável de decisão ociosidade, os modelos apresentaram alguns conflitos em relação ao *makespan*. Este comportamento ocorreu em virtude do ambiente utilizado por Gómez possuir o tempo ocioso somente ao final do turno. Para o ambiente de JSSP utilizado no modelo proposto, a ociosidade do turno ocorre em dois momentos: (a) no decorrer do turno, pois a máquina pode estar esperando pela parte da máquina anterior, e (b) ao final do turno, devido o tempo de processamento do lote ser maior que o tempo restante do turno. Outro fator importante está na relação da variável ociosidade com a variável parada, ou seja, se o tempo total ocioso não baixar significativamente e o tempo total de paradas aumentar, conseqüentemente o *makespan* irá aumentar, pois as duas variáveis possuem relação direta no *makespan*.

Ao ser analisada a variável parada, é importante entender que o comportamento desta variável em relação à ociosidade, reflete também a flexibilidade do sistema em trabalhar com o tamanho dos lotes de produção diversificados, ou seja, quanto maior o tamanho de lotes, menor o número de paradas de produção e maior o tempo ocioso.

Apesar dos modelos refletirem alguns comportamentos diferentes, ambos apresentaram o conflito entre a flexibilidade e a produtividade. Para o modelo proposto, a flexibilidade está relacionada com as datas de entrega, *mix* de partes, conjunto de máquinas e

rotas variadas, já a produtividade, com a velocidade de produção (otimização do *makespan*, tempo ocioso e tempo parado).

10 Conclusões

Neste trabalho foi proposto um modelo de escalonamento aplicado ao *Job-shop* num ambiente de Manufatura Flexível.

O ambiente de manufatura considerou um sistema de linha de produção com as rotas de partes variadas e um *mix* de partes com quantidades a produzir diversificadas. A escolha deste ambiente se deu através do alto grau de automação e da flexibilidade, característica da manufatura flexível.

O modelo implementado está baseado numa função objetivo que reflete as estratégias de escalonamento segundo as variáveis de decisão com seus pesos respectivos, e uma arquitetura que trata os problemas de seleção de partes e do escalonamento das partes.

A função objetivo é composta por quatro variáveis de decisão que buscam a otimização do tempo total de produção, tempo total de atraso, tempo total de paradas e o tempo total ocioso dos turnos. Para cada variável de decisão é considerado um peso. Já a arquitetura está dividida em cinco fases, onde: a primeira fase é responsável pela extração da demanda de produção, a segunda fase é responsável pelo agrupamento das partes em Famílias de Partes, a terceira fase é responsável pela geração da solução inicial através de regras de despacho, na quarta fase é obtida a otimização do escalonamento através do algoritmo Busca Tabu e, por fim, a quinta fase é responsável pela gravação do escalonamento num plano de produção.

10.1 Problema da Seleção de Partes e do Escalonamento

Dos problemas abordados e de interesse do modelo, são destacados os seguintes:

- O problema da Seleção de Partes, que é tratado na segunda fase da arquitetura, foi resolvido utilizando técnicas abordadas na Tecnologia de Grupo. Através da Análise por Fluxo de Produção e utilizando a técnica de formulação matricial, foi possível agrupar as partes a serem escalonadas através de similaridades de processo, neste caso, compartilhamento entre máquinas e operações. O objetivo deste agrupamento é o de evitar parada de produção e o melhor gerenciamento das partes, visto que, estas são tratadas em *clusters* e não na totalidade das partes. O Algoritmo utilizado foi proposto por Kusiak e é denominado de *Cluster Identification*.

- O problema do Escalonamento das partes, que é considerado na quarta fase, busca o seqüenciamento das partes de modo a respeitar as datas de entrega e a capacidade fabril (turnos de produção e máquinas). De acordo com os trabalhos relacionados mostrou que, em virtude da complexidade deste problema, estimula-se o uso das meta-heurísticas. A meta-heurística utilizada foi a Busca Tabu que, a partir de uma solução inicial viável, é responsável pela otimização do escalonamento. O algoritmo realiza a otimização do escalonamento a partir da função objetivo que, através das suas variáveis de decisão, reflete uma estratégia de escalonamento. Além da função objetivo, o algoritmo Busca Tabu utilizou dois critérios para a geração de vizinhança, onde: o primeiro critério está baseado nos movimentos de troca de dois lotes, e o segundo critério, consiste na retirada e inserção da parte de um lote. O primeiro critério de movimento tem como objetivo a redução do número de paradas de produção, visto que, a troca entre lotes possibilita a união de lotes da mesma Família de Partes. Já o segundo critério de vizinhança baseia-se na redução dos atrasos, dado que a retirada e inserção da parte está condicionada tal que o tempo de atraso fosse reduzido.

10.2 Resultados dos experimentos e validação

Através do estudo de caso, foi possível realizar diversos experimentos nas quais os pesos para cada variável de decisão da função objetivo foram variados de forma a otimizar: o tempo total de produção (*makespan*), o tempo total de atraso, o tempo total de paradas e o tempo total de ociosidade. As análises realizadas consideram a solução real de produção, a solução não-tendenciosa e a solução tendenciosa. A solução real de produção foi obtida através dos dados reais do estudo de caso, sendo apenas calculada a função objetivo, ou seja, o escalonamento extraído de produção não sofreu otimização. Já a solução não-tendenciosa foi obtida através da solução inicial (até a terceira fase da arquitetura – estudo de caso), de modo que todas as variáveis de decisão tivessem a mesma contribuição na função objetivo. A solução tendenciosa reflete as diversas políticas de otimização dada a variação do peso para cada variável de decisão.

Conclui-se que, através da estratégia de otimização que privilegiou o *makespan* deste estudo de caso, ficou evidente a relação entre o tempo total de parada e o tempo total de ociosidade, dado que, aumentando o *makespan*, ambas as variáveis aumentaram e, na redução, ambas as variáveis diminuíram. Contudo, em relação à variável de decisão atraso, demonstrou que o conflito entre a produtividade e as datas de entrega ocorreu. Se, por um lado, busca-se a

produtividade da produção através da otimização do *makespan*, por outro, perde-se a capacidade de entrega das partes no prazo solicitado pelo cliente, dado que não exista relação entre o *makespan* e o atraso.

Em relação à estratégia que privilegiou a variável de decisão atraso, foi verificado o comportamento inverso à variável de decisão *makespan*. Ao beneficiar o peso para o tempo total de atraso, foi constatado o aumento significativo do *makespan*, do tempo total de paradas e, aumento pouco significativo do tempo total da ociosidade. Este comportamento reflete à entrega pontual do cliente, porém, com baixa produtividade. Através da solução real de produção, foi constatado este comportamento, visto que, o escalonamento extraído do estudo de caso buscou a entrega pontual das partes e, conseqüentemente, o aumento dos instantes de paradas.

A partir da estratégia que beneficia a variável de decisão parada, demonstrou-se que, a dependência em relação às variáveis *makespan* e ociosidade ocorreram. Contudo, não refletiu os mesmos resultados obtidos pela função objetivo quando considerado o *makespan*. Este comportamento aconteceu em virtude da magnitude dos pesos de cada variável de decisão, dado que, ao aumentar o peso para a variável parada, as demais tornaram-se menos significativas na busca no espaço. Outro fator que demonstra este comportamento está na baixa dependência desta variável em relação às variáveis *makespan* e ociosidade.

A estratégia que privilegia a variável de decisão ociosidade apresentou um comportamento conflitivo em relação à variável parada. Este comportamento é conseqüência da pouca influência da variável parada dado o aumento do peso da variável ociosidade. Embora a variável *makespan* tenha minimizado, isto não reflete a relação de dependência destas variáveis, visto que a variável parada não aumentou significativamente. Caso o tempo parado aumentasse mais rapidamente do que o tempo ocioso, o *makespan* aumentaria também, pois ambas as variáveis parada e ociosidade, contribuem para a otimização do *makespan*. A utilização desta estratégia deve ser feita juntamente com as variáveis *makespan* e parada, de modo a utilizar o peso que não cause conflito. Já a variável atraso não refletiu com esta estratégia, concluindo assim que, o atraso e a ociosidade são variáveis independentes. Em outros ambientes de manufatura, a variável ociosidade pode sugerir a seguinte conclusão: (a) quanto maior o tempo de ociosidade, maior o tamanho dos lotes e, conseqüentemente, maior o *makespan* e menor o tempo total de paradas, (b) quanto menor o tempo de ociosidade, menor o tamanho dos lotes de produção, conseqüentemente, menor o *makespan* e maior o tempo total de paradas.

Considerando a variação do nbmax e o tamanho da lista tabu, concluiu-se que a variável de decisão atraso é privilegiada dado que o nbmax aumentou a política de intensificação do algoritmo BT e a lista tabu, refletiu no caráter de diversificação, resultando assim na exploração de regiões de busca até o momento desconhecidas.

No entanto, com a realização de diversos experimentos, foi possível verificar que várias estratégias obtiveram um escalonamento melhor do que a solução real utilizada pelo estudo de caso, mostrando assim a qualidade do modelo proposto. Dessa forma, a escolha de uma estratégia de escalonamento reflete diretamente com as estratégias que a produção deseja atingir, ou seja, o gerenciamento entre a entrega pontual e a alta produtividade.

Através da validação, foi possível concluir que o modelo possui integridade ao ser comparado com outras abordagens do JSSP (análise quantitativa) e com o MSPF (análise qualitativa). A Tabela 10.1 ilustra os resultados dos experimentos.

Tabela 10.1 – Resultados dos experimentos

Instância	Dimensão	SOC	MP	Gonçalves 2005	Aiex 2003	Wang 2001	Nowicki 1996	Dorndorf 1995	Aarts 1994	Storer 1992
FT06	6x6	55	55	55	55	55	55			
FT10	10x10	930	930	930	930	930	930	960	935	952
FT20	20x5	1165	1165	1165	1165	1165	1165	1249	1165	
LA01	10x5	666	666	666	666	666	666	666	666	666
LA06	15x5	926	926	926	926	926	926	926	926	
LA11	20x5	1222	1222	1222	1222	1222	1222	1222	1222	
LA16	10x10	945	945	945	945	945	945	1008	977	981
LA21	15x10	1046	1047	1046	1057	1058	1047	1139	1084	
LA29	20x10	1157	1160	1196	1203		1160	1336	1290	
LA40	15x15	1222	1229	1241	1244		1229	1321	1273	1258

Dessa forma, além de obter os valores ótimos conhecidos para os problemas supracitados na Tabela 10.1, o modelo proposto teve como principal contribuição o gerenciamento entre as datas de entrega e a produtividade do ambiente de produção, visto que as demais abordagens tratam somente o *makespan*. As Figuras 10.1 e 10.2 ilustram o comparativo do modelo proposto (MP) em relação à solução ótima conhecida (SOC) e as demais abordagens, para problemas de pequena e média escala.

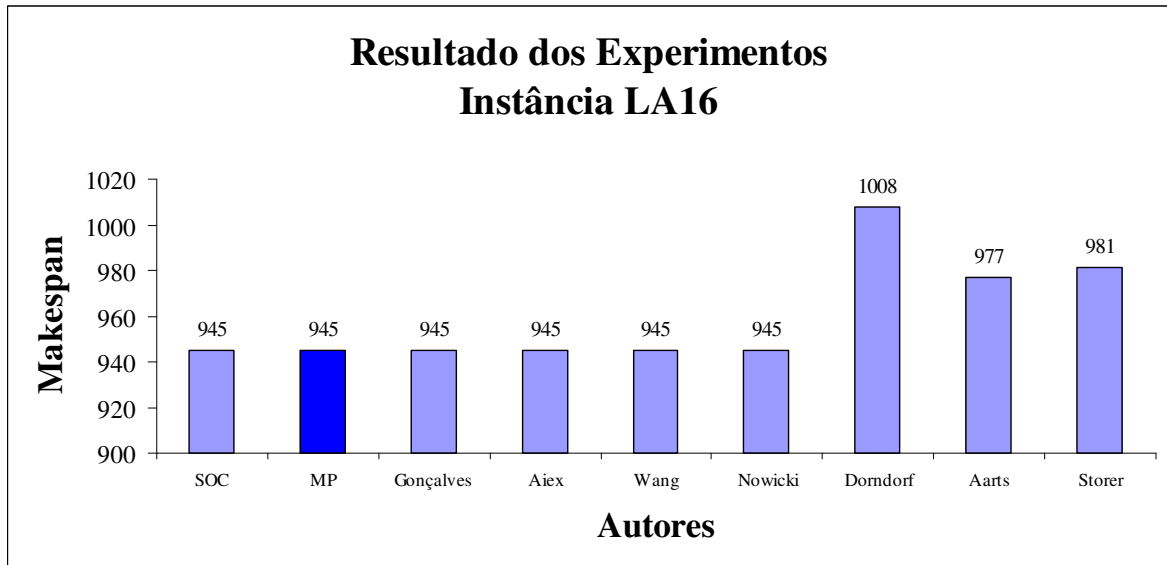


Figura 10.1 - Gráfico comparativo instância LA16 (pequena escala).



Figura 10.2 - Gráfico comparativo instância LA40 (média escala).

10.3 Características do Ambiente de Manufatura

Das diversas abordagens para o JSSP, este trabalho possibilitou a geração de um modelo aplicado em ambientes de manufatura com as seguintes características de escalonamento:

- o ambiente de produção é formado por “m” máquinas, “n” jobs e “i” operações (*Job-shop*);
- cada parte possui uma rota padrão (restrição para o *Job-shop*);
- uma máquina pode somente processar uma parte por vez (restrição para o *Job-shop*);

- uma parte pode somente ser processada em uma máquina por vez (restrição para o *Job-shop*);
- não existe quebra de rota (restrição para o *Job-shop*);
- o processamento de uma parte na máquina não pode ser interrompido, ou seja, processamento não-preemptivo (restrição para o *Job-shop*);
- as rotas são variadas (característica da Manufatura Flexível);
- as partes são diversificadas, ou seja, *mix* de partes (característica da Manufatura Flexível);
- os lotes são diversificados (característica da Manufatura Flexível); e
- capacidade fabril é respeitada (máquinas e turnos de produção).

Todas estas características de modo a gerenciar o conflito entre a flexibilidade, que está relacionada com as datas de entrega, *mix* de partes, conjunto de máquinas e rotas variadas, e a produtividade, relacionada com a velocidade de produção (otimização do *makespan*, tempo ocioso e tempo parado).

10.4 Trabalhos futuros

A seguir, são apresentados alguns trabalhos que poderão ser realizados como continuidade desta dissertação:

- Primeiro momento
 - calibração dos pesos para cada variável de decisão de forma a ajustar o modelo automaticamente;
 - aprimorar as técnicas de geração de vizinhança;
 - utilizar outras regras de despacho juntamente com as Famílias de Partes de modo a encontrar uma solução inicial melhor;
 - modificar o algoritmo BT para a utilização de somente uma lista tabu; e
 - aprimorar o método de geração da FP de acordo com as especificidades da produção.
- Segundo momento
 - propor e desenvolver instâncias do problema para o *Job-shop* de modo a incluir as datas de entrega e paradas de produção;
 - utilizar outros algoritmos para o escalonamento; e
 - montar um *Framework* da arquitetura.

Referências Bibliográficas

- Aarts, E.H.L.; Laarhoven, V.; P.J.M.; Lenstra, J.K.; Ulder, N.L.J. **A computational study of local search algorithms for job shop scheduling**, ORSA Journal on Computing 6, 1994, 118–125.
- Aiex, R.M.; Binato, S.; Resende, M.G.C.. **Parallel GRASP with path-relinking for job shop scheduling**, Parallel Computin, 2003, 29, 393–430.
- Arzi, Y.; Bukchin, J.; Masin, M. **An efficiency frontier approach for the design of cellular manufacturing systems in a lumpy demand environment**, European Journal of Operational Research, 2001, 134, 346-364.
- Askin, Ronald G.; Standridge, Charles R. **Modeling and Analysis of Manufacturing Systems**, John Wiley & Sons, Inc., 1993, 461p.
- Bedworth, David D.; Henderson, Mark R.; Wolfe Philip M. **Computer-Integrated Design and Manufacturing**, Mc-Graw-Hill, 1991, 653p.
- Blazewicz, Jacek; Domschke, Wolfgang; Pesch, Erwin. **The job shop scheduling problem: Conventional and new solution techniques**, European Journal of Operational Research, 1996, 93, 1-33.
- Blazewicz, Jacek; Ecker, H. Klaus; Pesch, Erwin; Schmidt, Gunter; Weglarz, Jan. **Scheduling Computer and Manufacturing Processes**, Springer, 2002, 485p.
- Blum, C.; Roli, A. **Metaheuristics in combinatorial optimization: overview and conceptual comparison**, Technical Report, TR/IRIDIA/2001-13, IRIDIA, Université Libre de Bruxelles, 2001.
- Browne, J.; Dubois D.; Rathmill K.; Sethi S.P.; Stecke, K.E. **Classification of Flexible Manufacturing Systems**, FMS Magazine, 1984, Vol. 2, 114-117.
- Burbidge, John L. **Production Flow Analysis**, The Production Engineer, 1971, v.50, n. 4/5, 139-142.
- Chandrasekharan, M. P.; Rajagopalan, R. **Groupability: Analysis of the properties of binary data matrices for group technology**, International Journal of Production Research, 1989, 27(6), 1035-1052.
- Chen, J.; Heragu, S.S. **Stepwise decomposition approaches for large scale cell formation problems**, European Journal of Operational Research, 1999, 113, 64-79.
- Choobineh, F. **A framework for the design of cellular manufacturing systems**, International Journal of Production Research, 1988, 26(7), 1161–1172.

Dorndorf, U.; Pesch, E. **Evolution based learning in a job shop environment**, Computers and Operations Research, 1995, 22, 25–40.

Garey, M. R.; Johnson, D.S.; Sethi, R. **The complexity of flowshop and jobshop scheduling**, Mathematics of Operations Research 1, 1976, 117-129.

Garey, M. R.; Johnson, D. S. **Computers and intractability: a guide to the theory of NP-completeness**, W. H. Freeman and Company, 1979, 338p.

Gen, Mitsuo; Cheng, Runwei. **Genetic Algorithms and Engineering Design**, Wiley-IEEE, 1996, 432p.

Gendreau, M. et al. **A Tabu Search Heuristic for the Vehicle Routing Problem**, Management Science, 1994, Vol. 40, 1276-1290.

Glover, Fred; Laguna, Manuel. **Tabu Search**, Kluwer Academic Publishers, 1997, 382p.

Gómez, Arthur T. **Seqüenciamento de Partes e Horários em um Sistema de Manufatura Flexível composto de uma máquina com restrições de ferramentas**, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, São Paulo, 1996.

Gonçalves, José Fernando; Resende, Mauricio G. C. **An evolutionary algorithm for manufacturing cell formation**, Computers & Industrial Engineering 47, 2004, 247-273.

Gonçalves, José Fernando; Mendes, J. J. M; Resende, Maurício G. C. **A hybrid genetic algorithm for the job shop scheduling problem**, European Journal of Operational Research, 2005, 167, 77-95.

Groover, Mikell P. **Automation, Production Systems, and Computer-Integrated Manufacturing**, Prentice Hall, 2001, Segunda Edição, 856p.

Hoffmann, Leandro T.; Rodrigues, Antonio G.; Gómez, Arthur T. **Análise de Métodos de Geração de vizinhanças em Pesquisa Tabu Aplicada em um Sistema de Manufatura Flexível**, XIII Salão de Iniciação Científica, Universidade Federal do Rio Grande do Sul, 2001.

Hurink, Johann; Knust, Sigrid. **Tabu Search algorithms for job-shop problems with a single transport robot**, European Journal of Operational Research, 2004.

Irani, Shahrukh, A.; Subramanian, Sanjay; Allam, S. Yosef. **Handbook of Cellular Manufacturing Systems**, Wiley-IEEE, 1999, 776p.

Jain, A.S.; Meeran, S. **Deterministic job-shop scheduling: Past, present and future**, European Journal of Operational Research, 1998, 113(2), 390-434.

Jha, Nand K. **Handbook of Flexible Manufacturing Systems**, Academic Press Limited, 1991, 328p.

Kandiller, Levent. **A cell formation algorithm: Hypergraph approximation – Cut tree**, European Journal of Operational Research, 1998, 109, 686-702.

King, J. R.; Nakornchai, V. **Machine-component group formation in group technology: Review and extension**, International Journal of Production Research, 1982, 20(2), 117–133.

Kumar K. R.; Chandrashekharan, M. P. **Grouping Efficacy: A quantitative criterion for goodness of block diagonal forms of binary matrices un group technology**, International Journal of Production Research, 1990, 28(2), 233-243.

Kusiak, A.; Chow, W. **Efficient solving of the group technology problem**, Journal of Manufacturing Systems, 1987, 6(2), 117–124.

Kusiak, A. **The generalized group technology concept**, International Journal of Production Research, 1987b, 25(4), 561–569.

Kusiak, A. **Intelligent Manufacturing Systems**, Prentice Hall, Englewood Cliffs, New Jersey, 1990, 443p.

Kusiak, A. **Intelligent Design and Manufacturing**, John Wiley & Sons, Inc., 1992, 753p.

Kusiak, A.; Dorf, Richard C. **Handbook of Design, Manufacturing and Automation**, John Wiley & Sons, Inc., 1994, 1042p.

Lawrence, S. **Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques**, Graduate school of industrial administration, Carnegie Mellon University: Pittsburgh, 1984.

Leung, Joseph Y-T. **Handbook of Scheduling: Algorithms, Models, and Performance Analysis**, CRC Press, 2004, 1224p.

Lorini, Flávio José. **Tecnologia de Grupo e organização da manufatura**, Editora da UFSC, 1993, 105p.

Mascis, A.; Pacciarelli, D. **Job-shop scheduling with blocking and no-wait constraints**, European Journal of Operational Research, 2002, 143, 498-517.

Mood, A. M.; Graybill, F.A.; Boes, D.C. **Introduction to the theory of statistics**, McGraw-Hill, 1974, 564p.

Muth, J. F.; Thompson, G. L. **Industrial Scheduling**, Englewood Cliffs, Prentice Hall, 1963, 225-251.

Nsakanda, A. L.; Diaby, M.; Price, L. W. **Hybrid genetic approach for solving large-scale capacitated cell formation problems with multiple routings**, European Journal of Operational Research, 2006, 171, 1051-1070.

Nowicki, E.; C. Smutnicki. **A Fast Taboo Search Algorithm for the Job Shop**, Management Science 42, 1996, 797-813.

Papadimitriou, Christos H. **Computacional complexity**, Reading: Addison-Wesley, 1995, 523p.

Rajagopalan, R.; Batra, J. L. **Design of cellular production systems: a graph-theoretic approach**, International Journal of Production Research, 1975, 13(6), 567–579.

Rantakyro, L. **Strategic management in small metal job shops in Sweden and the U.S.**, Engineering Management Journal, 2000, 12(2), 15-23.

Reeves, C.R. **Improving the efficiency of tabu search in machine sequencing problems**, J.Opl.Res.Soc., 1993, 44, 375-382.

Rogers, D. F.; Kulkarni, S. S. **Optimal bivariate clustering and a genetic algorithm with an application in cellular manufacturing**, European Journal of Operational Research, 2005, 160, 423-444.

Slack, Nigel et al. **Administração da Produção**, Atlas, 1997, 726p.

Storer, R. H.; Wu, S.D.; Parks, I. **Genetic algorithms in problem space for sequencing problems**, Proceedings of a Joint US-German Conference on Operations Research in Production Planning and Control, 1992, 584-597.

Tempelmeier, Horst; Kuhn, Heinrich. **Flexible Manufacturing Systems: Decision Support for Design and Operation**, Wiley-IEEE, 1993, 488p.

Tsai, Ying-Wei; Gemmill, Douglas D. **Using tabu search to Schedule activities of stochastic resource-constrained projects**, European Journal of Operational Research, 1998, 111, 129-141.

Vannelli, A.; Kumar, K. R. **A method for finding minimal bottleneck cells for grouping part-machine families**, International Journal of Production Research, 1986, 24(2), 387–400.

Viana, Geraldo Valdisio Rodrigues. **Meta-heurísticas e programação paralela em otimização combinatória**, EUFC, 1998, 250p.

Wang, L.; Zheng, D. **An effective hybrid optimization strategy for job-shop scheduling problems**, Computers & Operations Research, 2001, 28, 585-596.

Widmer, M. **Job Shop Scheduling with Tooling Constraints: a Tabu Search Approach**, Operational Research Society, 1991, 42, 75-82.

Wu, Bin. **Manufacturing Systems Design and Analysis**, Springer, 1994, 484p.

Xambre, A. R.; Vilarinho, P. M. **A simulated annealing approach for manufacturing cell formation with multiple identical machines**, 2003, 151, 434-446.

Zoghby, Jeriad; Barnes, Wesley L.; Hasenbein, John J. **Modeling the Reentrant job shop problem with setups for metaheuristic searches**, European Journal of Operational Research, 2004, 167, 336-348.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)