

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

APLICAÇÃO DO ALGORITMO GENÉTICO NA FORMAÇÃO
DE CÉLULA DE MANUFATURA

TESE APRESENTADA À
UNIVERSIDADE FEDERAL DE SANTA CATARINA
PARA A OBTENÇÃO DO GRAU
DE DOUTOR EM ENGENHARIA MECÂNICA

POR

VERA LUCIA DUARTE DO VALLE PEREIRA



0.234.200-4

UFSC-BU

FLORIANÓPOLIS, SANTA CATARINA

MARÇO, 1995

i

APLICAÇÃO DO ALGORITMO GENÉTICO NA FORMAÇÃO DE CÉLULA DE MANUFATURA

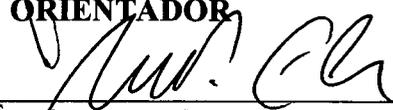
VERA LUCIA DUARTE DO VALLE PEREIRA

**ESTA TESE FOI JULGADA ADEQUADA PARA OBTENÇÃO DO TÍTULO DE
DOUTOR EM ENGENHARIA**

**ESPECIALIDADE ENGENHARIA MECÂNICA, ÁREA DE CONHECIMENTO
SÓLIDO E APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-
GRADUAÇÃO**



PROF.. ABELARDO ALVES DE QUEIROZ, Ph.D.
ORIENTADOR

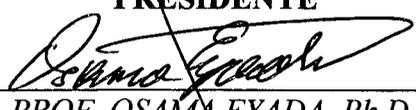


PROF. ANTÔNIO FÁBIO C. SILVA, Dr.Eng.
COORDENADOR DO CURSO

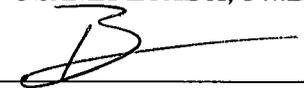
BANCA EXAMINADORA



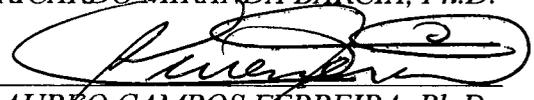
PROF.. ABELARDO ALVES de QUEIROZ, Ph.D.
PRESIDENTE



PROF. OSAMA EYADA, Ph.D.



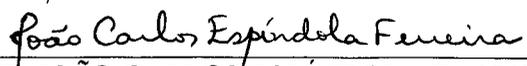
PROF. RICARDO MIRANDA BARCIA, Ph.D.



PROF. AUREO CAMPOS FERREIRA, Ph.D.



PROF. EDUARDO VILLA GONÇALVES FILHO, Ph.D.



PROF. JOÃO CARLOS ESPÍNDOLA FERREIRA, Ph.D.

*Ao Meu Esposo, que caminhou junto
nesta jornada apoiando constantemente
a realização deste trabalho.
Aos meus filhos Carmem, Christine,
Leandro e Gustavo e o meu genro Regis
pelo carinho e compreensão.*

Aos meus pais Carmen e Francisco.

AGRADECIMENTOS

Ao Professor Osama Eyada pela orientação prestada e o constante apoio e interesse na elaboração deste trabalho.

Ao Professor Abelardo Alves de Queiroz pela sua orientação e pelas valiosas contribuições.

Ao Professor Ricardo Miranda Barcia pelo constante apoio e incentivo.

Ao doutorando Antônio Carlos da Silva pela paciência nas horas difíceis do ajuste do programa de computação.

Aos professores e funcionários do Departamento de Engenharia de Produção e Sistemas pelo constante apoio e amizade transmitida ao longo do tempo.

Ao Josué Farias Dal Degan pelo incansável trabalho de diagramação.

E a todos que direta ou indiretamente contribuíram para a realização desse trabalho.

SUMÁRIO

AGRADECIMENTOS	IV
SUMÁRIO	V
LISTA DE FIGURAS	VII
LISTA DE TABELAS	IX
LISTA DE ABREVIACÕES	X
GLOSSÁRIO	XI
RESUMO	XIV
ABSTRACT	XV
CAPÍTULO 1	1
1. - IMPORTÂNCIA DA PESQUISA	1
1.1 Definição do Problema	2
1.2 Trabalhos Desenvolvidos	3
1.3 Necessidade da Pesquisa	4
1.4 Objetivo da Pesquisa	6
1.5 Organização do Trabalho	7
CAPÍTULO 2	7
2. - Tecnologia de Grupo	7
2.1 Manufatura Celular	10
2.2 Métodos para a Formação de Famílias de Peças	13
2.3 Métodos de Agrupamento	16
2.4 Técnicas baseadas em Arranjo	16
2.4.1 O método BEA (Bond Energy Analysis)	18
2.4.2 O método de Agrupamento ROC (Rank Order Clustering)	20
2.4.3 O Algoritmo de Agrupamento Direto (DCA)	22
2.5 Método de Análise de Agrupamento (Cluster analysis-based methods).	22
2.5.1 - Função Distância	23
2.5.2 Função Similaridade	25
2.6 Técnica de Reconhecimento de Padrão	31
2.7 Sistemas Especialistas	40
2.8 Análise de Agrupamento e Conjunto Difuso	41
2.9 Redes Neurais na Formação de Células de Manufatura	42
CAPÍTULO 3	45
3. - Algoritmo Genético	45
3.1 Uma Comparação entre o Algoritmo Genético e os Métodos Tradicionais	46
3.2 Aplicação Prática do Algoritmo Genético	49
3.3 Gabarito e sua Importância na Criação do Código	54
3.4 A Sobrevivência dos Gabaritos Mais Aptos	55
3.5 Aplicação Prática dos Teoremas	60
3.6 Uma retrospectiva sobre Algoritmo Genético	63
3.7 Conclusão	64

CAPÍTULO 4	65
4. - O Planejamento do Processo	65
4.1 Planos de Processos Alternativos	68
4.2 Agrupamento Máquina-Peça em Presença de Planos de Processo Alternativos	69
4.3 Conclusão	73
CAPÍTULO 5	74
5.- Modelo Proposto	74
5.1 Modelo Matemático	74
5.1.1 Notações	75
5.1.2 Modelo da Célula	76
5.2 Algoritmo Genético para a Otimização do Modelo Proposto	78
5.2.1 Processo Lógico de Formação do Algoritmo Genético	79
5.2.2 Criação do Código na Cadeia	80
5.2.3 Criar uma população	87
5.3 Considerações sob a Função Ajuste	89
CAPÍTULO 6	94
6. - Implantação do Modelo	94
6.1 Considerações Gerais	94
6.2 Código Computacional	95
6.3 Estrutura do Sistema	100
6.3.1 Criação de Novos Arquivos	102
6.3.2 Alterar Arquivo Existente	103
6.3.3 Rodar Arquivo	104
6.3.4 Apagar Arquivos	105
6.4 Criar um Problema para a Simulação	106
6.5 Determinação dos Parâmetros do Algoritmo Genético	114
6.6 Avaliação do desempenho	121
6.7 Conclusão	125
CAPÍTULO 7	126
7. - Análise dos Resultados	126
7.1 Considerações Gerais	126
7.2 Conclusões	128
7.3 Sugestões para a Continuação do Trabalho	128
Referências Bibliográficas	130
BIBLIOGRAFIAS	137
ANEXOS	153

LISTA DE FIGURAS

<i>Figura 2.1: Três tipos de arranjo físico: (a) linha; (b) grupo, (c) funcional</i>	9
<i>Figura 2.2 : Matriz inicial máquina-peça</i>	17
<i>Figura 2.3: Matriz final máquina-peça</i>	17
<i>Figura 2.4: Método BEA - rearranjo de matrizes</i>	19
<i>Figura 2.5: Método ROC - matriz inicial</i>	20
<i>Figura 2.6: Método ROC - Matriz intermediária</i>	21
<i>Figura 2.7 : Método ROC - matriz final</i>	21
<i>Figura 2.8: Matriz inicial máquina-peça</i>	27
<i>Figura 2.9: Matriz similaridade para a figura 2.8</i>	28
<i>Figura 2.10: Dendograma para os valores da figura 2.9</i>	29
<i>Figura 2.11: Classes de padrões num espaço bidimensional</i>	33
<i>Figura 2.12: Procedimento para classificação auxiliada por computador</i>	34
<i>Figura 2.13: Construção de um contorno</i>	35
<i>Figura 2.14: Conjunto de segmentos e um contorno</i>	36
<i>Figura 2.15: Diagrama das primitivas</i>	37
<i>Figura 2.16: Exemplo de padrão de reconhecimento</i>	38
<i>Figura 2.17: Representação de uma superfície</i>	39
<i>a) Sólido</i>	39
<i>b) Estrutura de Árvore</i>	39
<i>Figura 2.18: Representação matemática de um elemento em</i>	44
<i>processamento e suas conexões.</i>	44
<i>Figura 3.1: Gráfico da Função $f(x) = x^2$</i>	47
<i>Figura 3.2: Sistema de Chaves para Obtenção do Sinal $f(s)$</i>	47
<i>Figura 3.4 Roleta de Aptidões de Sobrevivência</i>	50
<i>Figura 3.5: Régua Gabarito</i>	54
<i>Figura 4.1: Fluxo de Informação do Planejamento do Processo</i>	66
<i>Figura 5.1: Representação de um cromossomo</i>	81
<i>Figura 5.2: Representação do cromossomo peça-grupo</i>	81
<i>Figura 5.3: Representação de um cromossomo peça-grupo-plano</i>	82
<i>Figura 5.4: Exemplo de um cromossomo peça-grupo-plano</i>	82

<i>Figura 5.5: Representação do cromossomo máquina</i>	82
<i>Figura 5.6: Representação do cromossomo máquina-grupo</i>	83
<i>Figura 5.7: Exemplo de um cromossomo máquina-grupo</i>	83
<i>Figura 5.8: Representação de um cromossomo completo segundo o código</i>	84
<i>Figura 5.9: Exemplo de cruzamento de Alelo</i>	85
<i>Figura 5.10: Representação do cromossomo efetivo</i>	85
<i>Figura 5.11: Exemplo de um par de cromossomo</i>	86
<i>Figura 5.12: Criação da Nova População</i>	88
<i>Figura 5.13: Representação de uma função com mínimo</i>	90
<i>Figura 5.14: Representação da Função Ajuste</i>	91
<i>Figura 5.15: Representação da Função $f(x) = x^2$</i>	92
<i>Figura 6.1 : Fluxo de Dados do Sistema</i>	101
<i>Figura 6.2 : Fluxograma de Lógica do Sistema "Criar Novo Arquivo".</i>	102
<i>Figura 6.3 : Fluxograma de lógica do sistema "alterar arquivo".</i>	103
<i>Figura 6.4 : Fluxograma de Lógica do Sistema "Rodar Arquivo".</i>	104
<i>Figura 6.4 (continuação) : Fluxograma de Lógica do Sistema "Rodar Arquivo".</i>	105
<i>Figura 6.5 : Fluxograma de Lógica do Sistema "Apagar Arquivo".</i>	106
<i>Figura 6.6 : Dados para Rodar o Algoritmo Genético - Exemplo Hipotético</i>	107
<i>Figura 6.7 :Curva de custo em relação ao número de gerações</i>	110
<i>Figura 6.8 :Curva de custo em relação ao número de gerações</i>	111
<i>Figura 6.9 :Curva de custo em relação ao número de gerações.</i>	112
<i>Figura 6.10 :Curva de custo em relação ao número de gerações.</i>	113
<i>Figura 6.11 : Resultado Ótimo - Exemplo Hipotético</i>	117
<i>Figura 6.12 : Matriz inicial máquina-peça</i>	123
<i>Figura 6.13 : Matriz Final</i>	123
<i>Figura 6.14 : Exemplo de uma solução computacional</i>	125

LISTA DE TABELAS

<i>Tabela 3.1: Valor da Função de Objetivo da Primeira Geração</i>	50
<i>Tabela 3.2: Exemplo de Aplicação do Algoritmo Genético</i>	52
<i>Tabela 3.3 : Criação da Nova População</i>	53
<i>Tabela 3.4 : Desempenho da Segunda Geração de Cadeias</i>	53
<i>Tabela 3.5: Formação dos Gabaritos</i>	61
<i>Tabela 3.6 :Gabaritos da Nova População</i>	61
<i>Tabela 6.1 : Planos de Processo e Ferramentas Requeridas pelas Peças</i>	115
<i>Tabela 6.2 : Ferramentas Disponíveis nas Máquinas</i>	116
<i>Tabela 6.3 : Número de Soluções Ótimas Usando Diferentes Gerações e Populações para Função Ajuste < 77000.00</i>	118
<i>Tabela 6.4 : Número de Soluções Ótimas Usando Diferentes Gerações e Populações para Função Ajuste < 78000.00</i>	118
<i>Tabela 6.5 : Número de Soluções Ótimas Usando Diferentes Gerações e Populações para Função Ajuste < 80000.00</i>	119
<i>Tabela 6.6 : Probabilidade de Cruzamento × Probabilidade de Mutação</i>	122
<i>Tabela 6.7 : Resultado computacional da figura 6.12</i>	124

LISTA DE ABREVIações

AG	<i>Algoritmo Genético</i>
ALCA	<i>Average Linkage Clustering Algorithm</i>
BEA	<i>Bond Energy Algorithm</i>
CAD	<i>Desenho Auxiliado por Computador</i>
CAM	<i>Fabricação Auxiliada por Computador</i>
CAPP	<i>Planejamento do Processo Auxiliado por Computador</i>
CIM	<i>Manufatura Integrada por Computador</i>
DCA	<i>Direct Clustering Algorithm</i>
FMC	<i>Célula Flexível de Manufatura</i>
FMS	<i>Sistema Flexível de Manufatura</i>
JIT	<i>Filosofia Just-in-Time</i>
PFA	<i>Análise do Fluxo da Produção</i>
ROC	<i>Rank Order Clustering</i>
SLCA	<i>Single-Linkage Clustering Algorithm</i>
TG	<i>Tecnologia de Grupo</i>

GLOSSÁRIO

Análise do Fluxo de Produção	Um procedimento estruturado para analisar a seqüência de operações que uma peça percorre durante a manufatura.
Arranjo Físico em Linha	As máquinas são dispostas na seqüência em que são usadas.
Average Linkage Clustering Alg.	Um algoritmo que agrupa coisas baseado na similaridade média de todos os pares de coisas.
Cad	É a criação e otimização do desenho usando o computador.
Célula	Um grupo de máquinas dispostas de maneira a produzir uma família de peças similares.
Classificação	O processo de categorização de peças em grupos, algumas vezes chamado de famílias, de acordo com o conjunto de regras ou princípios.
Cluster Analysis	O processo de classificar coisas em grupos de maneira que as similaridades são alta entre os membros do próprio grupo e baixa entre os membros de diferentes grupos.
Codificação	O processo de designar símbolos à uma peça para definir os seus atributos.
Código Híbrido	Uma combinação do monocódigo e do policódigo.
Coefficiente de Similaridade	Uma medida de como duas máquinas são semelhantes.
Computer-Aided Process Planning (Capp)	Um sistema computacional interativo que automatiza alguns dos trabalhos envolvidos na preparação da folha de processo.

Computer-Integrated Manufacturing (CIM)	Uma filosofia administrativa a qual as funções de projeto e manufatura são racionalizadas e coordenadas usando o computador , comunicação e as tecnologias de informações.
Dendograma	É a representação gráfica em forma de árvore dos resultados do cluster analysis.
Família de peças	Um grupo de peças que tem alguns atributos similares.
Flow shop	Sequenciamento onde o fluxo de peças obedece unidirecionalmente nos equipamentos necessários para seu processamento.
Job shop	Sequenciamento onde as peças não tem um sentido pré definido para seu processamento. Várias máquinas podem ser escolhidas para a mesma operação.
Just-in-Time	Uma filosofia administrativa que tem como meta a eliminação de qualquer coisa que não adiciona valor ao produto.
Lead time	Tempo total de processamento de uma determinada peça, desde a chegada em bruto até estar terminada a sua última como o de movimentação e de estoque intermediário.
Manufatura Celular (FMS)	A organização de máquinas e pessoas em grupos responsáveis pela produção de uma família de peças.
Máquina gargalo	Uma máquina no grupo que é requerida por um número grande de peças em diferentes grupos
Monocódigo (Hierárquico)	O significado de cada caracter é dependente do significado do caracter anterior.
Plano de processo	As instruções detalhadas de como fabricar uma peça.

Policódigo (Atributo)	O significado de cada caracter no código é independente de qualquer outro caracter.
Setup	Tempo de preparação de uma máquina para o processamento.
Single Linkage Clustering Algorithm	Um algoritmo de agrupamento de coisas com alto coeficiente de similaridade.
Inteligência Artificial	Projetado para simular a inteligência humana através de um software.
Tecnologia de Grupo	Designação de um código para uma peça a qual contem as características das peças.
Valor de Vigilância (Threshold)	Um valor do coeficiente de similaridade no qual o agrupamento para.

RESUMO

Neste trabalho estuda-se o problema de formação de células de fabricação. A formação de família de peças e grupos de máquinas é um problema que vem desafiando, nos últimos tempos, a argúcia dos pesquisadores. Este trabalho propõe um modelo matemático, com um procedimento de solução baseado no Algoritmo Genético abordando a questão do custo e tempo de processamento das peças.

A criação de problemas para teste mostraram que o Algoritmo tem solução e que ele é uma ferramenta valiosa para trabalhar no ambiente de fabricação.

ABSTRACT

In this work the problem of cell formation is studied. The machine cell and part families formation is a problem that has called the attention of research in Group Technology in the last years. This work proposes a mathematical model with a solution procedures based on Genetic Algorithm involving part cost and times processing.

The elaboration of sample problem trials suggest, that the proposed Algorithm can be solved and it is a powerful tool in cellular manufacturing environment.

CAPÍTULO 1

1. - IMPORTÂNCIA DA PESQUISA

As indústrias modernas têm-se defrontado com uma série de problemas oriundos do aumento da competitividade no mercado internacional como também de mudanças na tecnologia manufatureira obrigando-as a um melhor aproveitamento e racionalização dos recursos disponíveis de maneira a conseguir uma alta produtividade industrial. As indústrias do futuro deverão proporcionar processos mais flexíveis de maneira a otimizar os recursos tendo como resultado uma diminuição no "lead times", diminuição nos estoques, aumento da utilização dos equipamentos e aumento na qualidade e desempenho.

O principal propósito é a substituição da economia de escala de produção em massa por estratégias que facilitam a flexibilidade, reduzam o tempo do ciclo de projeto, do tempo de lançamento de um novo produto no mercado, e o ciclo de entrega do produto existente ao cliente (Browne, 1988).

A combinação entre a necessidade de diversificação de produto, a diminuição do ciclo de vida deste e a pressão para redução do tempo de entrega no mercado têm dado origem à necessidade de economia de escopo. Segundo Goldhar (1983), a economia de escopo existe "...onde o mesmo equipamento pode produzir mais economicamente peças de formas semelhantes pela combinação de ferramentas, do que produzi-las individualmente. Não faz a mínima diferença para uma máquina ferramenta controlada pelo computador produzir em série peças com projetos idênticos, ou, randomicamente, executar peças a serem realizadas sob diferentes projetos, desde que os projetos estejam limitados a uma família. Mudanças de tempo (conseqüentemente custos) são desprezíveis, visto que o tempo de preparação da máquina consiste somente em ler um programa de computador..." (Browne, Jimmie et al, 1988).

Em face da competitividade nacional e internacional cada vez mais crescente, mudanças urgentes deverão ser feitas pelos administradores forçando-os a tornar a organização mais produtiva de maneira a preencher as suas necessidades de sobrevivência, crescimento e prosperidade.

A Tecnologia de Grupo (TG) ou manufatura celular tem emergido como um fator fundamental na modernização industrial em aperfeiçoar a produtividade dos sistemas de manufatura em lotes. A redução dos custos dos computadores tem viabilizado o armazenamento de grande volume de dados.

A aplicação da Tecnologia de Grupo em conjunção com CAD e CAM pode fornecer um mecanismo poderoso para a integração dos sistemas de manufatura.

A manufatura celular viabiliza a produção de lotes de pequenos tamanhos com alta produtividade e baixo custo. A formação de células de manufatura baseia-se na classificação das peças e no agrupamento delas em famílias explorando as similaridades geométricas e/ou os processos.

Esta nova técnica (Tecnologia de Grupo) que agrupa peças em famílias e máquinas em células de manufatura, reconfigura o arranjo físico da fábrica de funcional para arranjo físico em grupo. Neste arranjo físico, o chão de fábrica é subdividido em células de manufatura, o qual contém um grupo de máquinas ferramentas e equipamentos, que processam um conjunto de famílias de peças. Estes grupos, devido a sua formação em pequenos lotes e em unidades diferentes de manufatura são capazes de produzir peças com maior eficácia.

Esta técnica de manufatura inovadora reduz o tempo de "setup", o de "throughput", material em processo, e o de percurso do material, conseqüentemente melhorando a qualidade e a satisfação do trabalhador (Kusiak,1987).

O surgimento destas novas técnicas não serão válidas se não houver um planejamento e uma implementação efetiva. A integração computacional é o fator predominante nas mudanças das fábricas. No passado, as fábricas davam ênfase aos processos físicos, já as do futuro visam basicamente a integração de dados.

A aplicação de computadores tem trazido efeito de racionalização nas indústrias mecânicas, provocando um aumento de produtividade e flexibilidade.

Do ponto de vista humano, o time que trabalha numa célula tem uma maior variabilidade de tarefas, pois opera os mais diferentes tipos de equipamentos dentro dela. Ele desenvolve com isto um maior senso de participação, pois o seu envolvimento com o processo produtivo, do início ao fim, auxilia-no na administração, planejamento e controle dos processos, verificando problemas e propondo soluções. Esta integração com o setor produtivo tem levado a um melhor desempenho, motivação e qualidade no trabalho.

1.1 Definição do Problema

Como já foi discutido por vários autores, existem vários métodos para solucionar o problema de agrupamento peça-máquina. O agrupamento de

peças e máquinas em células de maneira a termos uma família de peças usando um grupo de máquinas é o problema mais importante no estágio de planejamento de um sistema flexível de manufatura.

Existem vários métodos para solucionar o problema de agrupamentos, os quais pode podem ser enfocados de três maneiras: a) a primeira metodologia procura agrupar primeiramente as peças formando famílias das mesmas, seguindo da alocação das máquinas às famílias; b) a segunda procura agrupar as máquinas formando um grupo de máquinas, seguindo da alocação das peças ao grupo de máquinas; c) a terceira metodologia considera o problema do agrupamento simultâneo de peças e máquinas (Gunasingh & Lashkari, 1989; Rajamani,D. et al, 1990).

Na realidade, o número de peças é, na prática, maior de que o número de máquinas e alocar as máquinas às famílias de peças, é a solução mais fácil. Entretanto, em grandes sistemas de manufatura, a segunda metodologia é mais efetiva (Rajamani,D. et al, 1990). O terceiro método supõe dar melhores resultados, pois simultaneamente agrupa peças e máquinas. Contudo, ele apresenta menos atrativo em termos computacionais devido a necessidade de uma grande formulação matemática.

Outro problema apresentado na maioria dos modelos para o projeto de células de manufatura é a premissa de que para cada peça existe somente um plano de processo e que cada operação ou conjunto de operações são realizados somente por uma máquina . Ao se fixar uma máquina não ocorreria uma seleção ótima aumentando o custo de capital e a pouca utilização das mesmas, tornando-se uma desvantagem na manufatura celular.

Baseado nestas premissas, procurou resolver-se o problema de agrupamento simultâneo de peças e máquinas na presença de planos de processos alternativos através da apresentação de um modelo matemático, o qual procurou solucionar-se pelo método do Algoritmo Genético.

1.2 Trabalhos Desenvolvidos

O problema da formação de célula de manufatura tem atraído o esforço de um grande número de pesquisas e vários procedimentos foram desenvolvidos para solucioná-los. Nos primeiros anos de desenvolvimento, as informações para a formação de família de peças eram disponíveis nas folhas de processos das peças. Burbidge (1963, 1973, 1975) com seu trabalho pioneiro na formação de células de manufatura propôs uma técnica de análise do fluxo de produção anual baseado em folhas de processo. O seu trabalho era conveniente para pequenos problemas, razão porque sua metodologia foi aprimorada por

King (1980), Chan e Milner (1982), King e Nakornchai (1982), e Kusiak (1987). Estes algoritmos são baseados na permutação de linhas e colunas da matriz de incidência de maneira a ter ao redor da diagonal principal elementos diferentes de zero. Ao se rearranjar a matriz, permite ao projetista ver o agrupamento de máquinas-peças numa matriz efetivamente bem estruturada. Na maioria destes algoritmos, a solução final depende do arranjo da matriz inicial.

Uma metodologia analítica para a formulação e solução do problema de Tecnologia de Grupo foi proposta por McAuley(1972). O seu trabalho é conhecido como o método de coeficiente de similaridade (SCM). Ele computa este coeficiente de similaridade entre máquinas e componentes e procura agrupá-los usando análise de agrupamento ou dendograma. Seu método foi adotado posteriormente por Carrie (1973), De Witte (1979), Seifodini e Wolfe (1986).

Rajagopalan e Batra (1975) propuseram uma técnica para o problema de agrupamento máquina-componente através da introdução do conceito de teoria dos Grafos. A seleção de um valor limite (threshold) para o coeficiente de similaridade é arbitrária e necessita um certo grau de julgamento.

Algumas técnicas de decomposição heurísticas foram sugeridas também por Vanelli e Kumar (1986), e por Kumar et al (1986). Como foi apresentado anteriormente, achar a solução ótima no gráfico decomposição é uma tarefa combinatorial difícil e não garante a otimalidade.

Kusiak (1987) sugeriu um modelo para a formação de células de manufatura, o qual utiliza programação inteira, chamado de p-Median. Ele procura maximizar a soma dos coeficientes de similaridade para um número fixo de grupos, de maneira que a restrição de que cada peça pertença somente a uma família seja alcançada. Outros modelos matemáticos de programação inteira foram propostos por Choobinch (1988), Raja Gunasingh e Lashkari (1989).

Além destas metodologias, existem outras baseadas em técnicas recentes, como os sistemas especialistas (Kusiak, 1988), as quais serão discutidas com mais detalhes no capítulo 2.

1.3 Necessidade da Pesquisa

A maioria dos algoritmos apresentados são capazes de resolver o problema de Tecnologia de Grupo para matrizes bem estruturadas. Quando isto não acontece, eles falham e são incapazes de fornecer resultados aceitáveis. Além disto, os métodos tradicionais de agrupamento permitem somente o agrupamento de máquinas ou componentes, não alcançando o objetivo

primordial da formação de célula, que é o agrupamento simultâneo de máquinas e componentes.

Para solucionar o problema, procura-se através do conceito de Algoritmo Genético fazer o agrupamento simultâneo de peças e máquinas em presença de planos de processos alternativos¹. Nos aspectos computacionais informações relativas aos parâmetros de produção (tempo, custo, etc...), como também os parâmetros relativo ao Algoritmo Genético serão abordados.

Recentemente o conceito de Sistema Flexível de Manufatura (SFM) tem surgido como uma tendência primordial nas indústrias de manufatura. O objetivo básico dos Sistemas Flexíveis de Manufatura é alcançar a eficiência da produção em massa, enquanto conserva a versatilidade do "job shop". Entretanto, os Sistemas Flexíveis de Manufatura podem oferecer uma vantagem substancial em comparação com os sistemas de produção convencionais, e a realização destas vantagens requer a solução de problemas de planejamento e programação da produção. Estes problemas de planejamento e programação da produção apresentam questões complexas. para um Sistema Flexível de Manufatura real; é impossível resolver estas questões por um simples modelo analítico.

A Tecnologia de Grupo é a base racional para fornecer os meios de organização da manufatura celular, através das aplicações da filosofia, se obter uma série de vantagens operacionais e econômicas.

A manufatura celular é a divisão das máquinas da instalação de manufatura em células de produção. Cada célula é projetada para fazer uma família de peças, que é definida como um conjunto de peças que exigem máquinas, ferramental, operações de máquinas e/ou matrizes e dispositivos semelhantes. Normalmente, as peças dentro da família vão desde a matéria prima até as peças acabadas dentro de uma única célula.

Os efeitos econômicos da Tecnologia de Grupo são de grande relevância, mas é possível que a longo prazo seus efeitos sociológicos sejam de grande importância. A organização celular permite processar a matéria prima de seu estado inicial até atingir o produto acabado, com as tarefas sendo conduzidas por um pequeno número de trabalhadores, que podem acompanhar todas as fases e ver o resultado de seu empenho até o estágio final. Isto proporciona ao elemento humano um maior nível de satisfação, o de sentir os resultados de seu trabalho para as atividades da empresa, reduzindo drasticamente o "stress" e as chances de conflito.

Pelo fato de manter-se mais perto da atividade, integrado ao ambiente, o elemento humano sente-se mais responsável e pode detectar erros

¹ O termo alternativo neste trabalho se refere ao sentido clássico da palavra alternativo e não ao sentido que se tem dado em algumas áreas de conhecimento; "uma opção não convencional". Portanto aqui trata-se "de opções todas convencionais".

em condições de corrigi-los em tempo de processo, contribuindo para a melhoria da qualidade. São evidentes, portanto os ganhos em produtividade e qualidade, em confronto com os sistemas convencionais.

1.4 Objetivo da Pesquisa

Um dos mais importantes problemas no planejamento da manufatura celular é a formação da própria célula de manufatura. A identificação de famílias de peças e a formação de grupos de máquinas é, então, nosso problema. A manufatura celular é uma metodologia organizacional baseada no processamento de peças similares (família de peças) através de grupo de máquinas (células). A manufatura celular proporciona uma oportunidade para redução do tempo de "setup", de estoques intermediários e do processamento de uma determinada peça.

Nosso propósito é desenvolver uma metodologia que através do agrupamento de peças, as quais podem ser executadas sob diferentes planos de processo e utilizando um número pré definido de máquinas (inclusive repetidas), consigamos modelar a organização do processo de produção de modo mais eficiente e econômico.

O método proposto assume que uma peça pode ter mais de um plano de processo e que cada operação no plano de processo pode ser realizada em mais de uma máquina. O agrupamento de peças, para a formação de famílias, é baseado na capacidade das máquinas em processar essas peças, a qual se expressa em termos de ferramentas disponíveis em cada máquina. As exigências de processamento das peças estão relacionadas às seguintes especificações: à disponibilidade do ferramental, ao tempo de processamento, ao custo de produção para o par peça-máquina, e à capacidade de cada máquina.

A necessidade de desenvolvimento de um modelo que simule a realidade do meio de manufatura de maneira a aproveitar os benefícios da Tecnologia de Grupo foi a grande preocupação. Procurou-se através da formulação matemática representar um modelo que tivesse os seguintes objetivos:

- a) permitisse planos de processos alternativos e o bom balanceamento entre as operações realizadas numa célula;
- b) solução do modelo utilizando o Algoritmo Genético;
- c) a determinação dos valores apropriados dos parâmetros do Algoritmo Genético para a formação de célula de manufatura;

Além disso, será feita uma comparação dos resultados do modelo proposto com outros modelos apresentados por outros autores.

1.5 Organização do Trabalho

No capítulo 2 foi feita uma revisão dos métodos clássicos para a formação de família de peças. As técnicas mais recentes para a formação de células foram também apresentadas, como Reconhecimento de padrão, Sistemas Especialistas, Conjunto Difuso e Redes Neurais. Comentários sobre vantagens e desvantagens dos métodos também foram feitos.

No capítulo 3 foi feita uma explanação do Algoritmo Genético e as vantagens de sua utilização para a formação de células de manufatura. Algumas metodologias usadas no planejamento do processo são apresentadas no capítulo 4.

O modelo proposto é apresentado no capítulo 5. A implantação do modelo com a ilustração usando um exemplo para o agrupamento de máquinas e peças é apresentado no capítulo 6. Algumas discussões dos parâmetros empregados na formação de famílias de peças utilizados pelo Algoritmo Genético e extensões do modelo, conclusões com sugestões são apresentadas no último capítulo.

CAPÍTULO 2

2. - Tecnologia de Grupo

O conceito de Tecnologia de Grupo (TG) foi primeiramente discutido por Mitrofanov em Leningrado na União Soviética, cerca de 40 anos atrás. Infelizmente não foi largamente adotado até o início de 1970. Desde então, as Tecnologias Auxiliadas pelo Computador e os Sistemas Flexíveis de Manufatura (FMS) foram introduzidos nas indústrias, e a Tecnologia de Grupo foi um dos fatores que levou ao sucesso dos projetos dos Sistemas Flexíveis de Manufatura (ElMaraghy, Hoda A. et al, 1988).

O conceito de Tecnologia de Grupo é fundamentalmente simples, mas com um grande potencial. Basicamente é uma metodologia de organização, onde temos sistemas de fabricação em pequenos e médios lotes com grande variedade de peças. Estes princípios e os benefícios oriundos da aplicação de TG têm sido aplicados em vários tipos de indústrias manufatureiras, com maior ênfase em peças mecânicas.

A definição de Tecnologia de Grupo é a seguinte:

"É uma filosofia de produção para pequenos e médios lotes de peças, onde as mesmas são identificadas e agrupadas juntas para se recolherem as vantagens de suas similaridades em processos e/ou projetos, em todos os setores da organização"(Mitrofanov, 1966).

As peças deixam de ser partes autônomas passando a pertencer a uma família com características bem definidas.

A idéia primordial de Tecnologia de Grupo é o agrupamento de peças com similaridades de processos, em famílias de peças formando uma célula de fabricação, e que uma única solução pode ser encontrada para um conjunto de problemas, poupando assim tempo e esforços. Definimos família de peças como uma coleção de peças, as quais podem ser produzidas usando o mesmo conjunto de recursos. A Tecnologia de Grupo (TG) é utilizada na manufatura para identificar família de peças, as quais podem ser fabricadas por processos similares. Estas peças não serão processadas individualmente, mas em conjuntos de peças similares, dentro de uma determinada seqüência, reduzindo os tempos improdutivos e os custos de manufatura, obtendo com isto grandes vantagens econômicas devido às suas similaridades (Groover, M., 1987). Este procedimento conduz ao agrupamento de máquinas para processar estas famílias. Estes agrupamentos de máquinas são normalmente conhecidos por células de fabricação. Outra premissa é de que pelo aproveitamento dos dados

de desenhos existentes, para o projeto de novas peças com similaridades geométricas, deram origem à padronização das peças, reduzindo com isto tempo e esforço, se esta informação já for disponível. Estes são alguns dos motivos da utilização de Tecnologia de Grupo como uma filosofia de fabricação pela maioria da comunidade manufatureira, com resultados significativos na melhoria da qualidade e produtividade.

A aplicação de Tecnologia de Grupo na produção pode ser viabilizada pelo sequenciamento de uma família de peças similares através de um arranjo físico funcional ou produzindo as famílias destas num grupo especializado de máquinas. Logo, o tipo de arranjo físico para máquina e equipamento assume um papel primordial não somente em função dos processos, como também na racionalização dos tempos de passagem entre eles. Normalmente, as indústrias que trabalham com médios lotes utilizam o arranjo físico funcional, enquanto o arranjo físico em linha é para produção em série.

Nos sistemas industriais existem basicamente três tipos de "arranjo físico": produto ou linha, processo ou funcional e celular ou grupo.

O arranjo físico de produto ou arranjo físico em linha é o arranjo físico típico de produção em massa, isto é, as máquinas são arranjadas na seqüência das operações da fabricação contínua. Este arranjo físico se caracteriza pela grande produtividade e baixa flexibilidade.

No arranjo físico funcional as máquinas são arranjadas de acordo com a sua função, ou seja, as máquinas do mesmo tipo são colocadas juntas, como torno com torno, fresa com fresa, etc.. É um arranjo típico de especialização por processo, onde as máquinas que realizam processos semelhantes são agrupadas na mesma área física.

O arranjo físico funcional é usado onde é necessário um alto grau de flexibilidade e baixo volume de produção. Peças com mais de um processo se movimentam de uma seção para outra até o término da fabricação. O arranjo físico funcional tem a desvantagem do aumento de custo através de uma grande movimentação de material, uma grande quantidade de estoques intermediários, excessivo tempo de "setup" e perdas de vendas.

O arranjo físico celular ou de grupo consiste em agrupar pessoas e máquinas em uma área especialmente montada para isto, de maneira a produzir uma ou mais famílias de peças. O arranjo físico celular é fácil de ser administrado, pois a capacidade de uma célula é determinada pelas famílias de peças que utilizam a célula. Este tipo de arranjo físico traz uma série de benefícios, como na melhoria do controle de produção, redução do tempo de preparação e fabricação, etc..., e proporciona os fundamentos para o desenvolvimento de uma célula flexível automatizada.

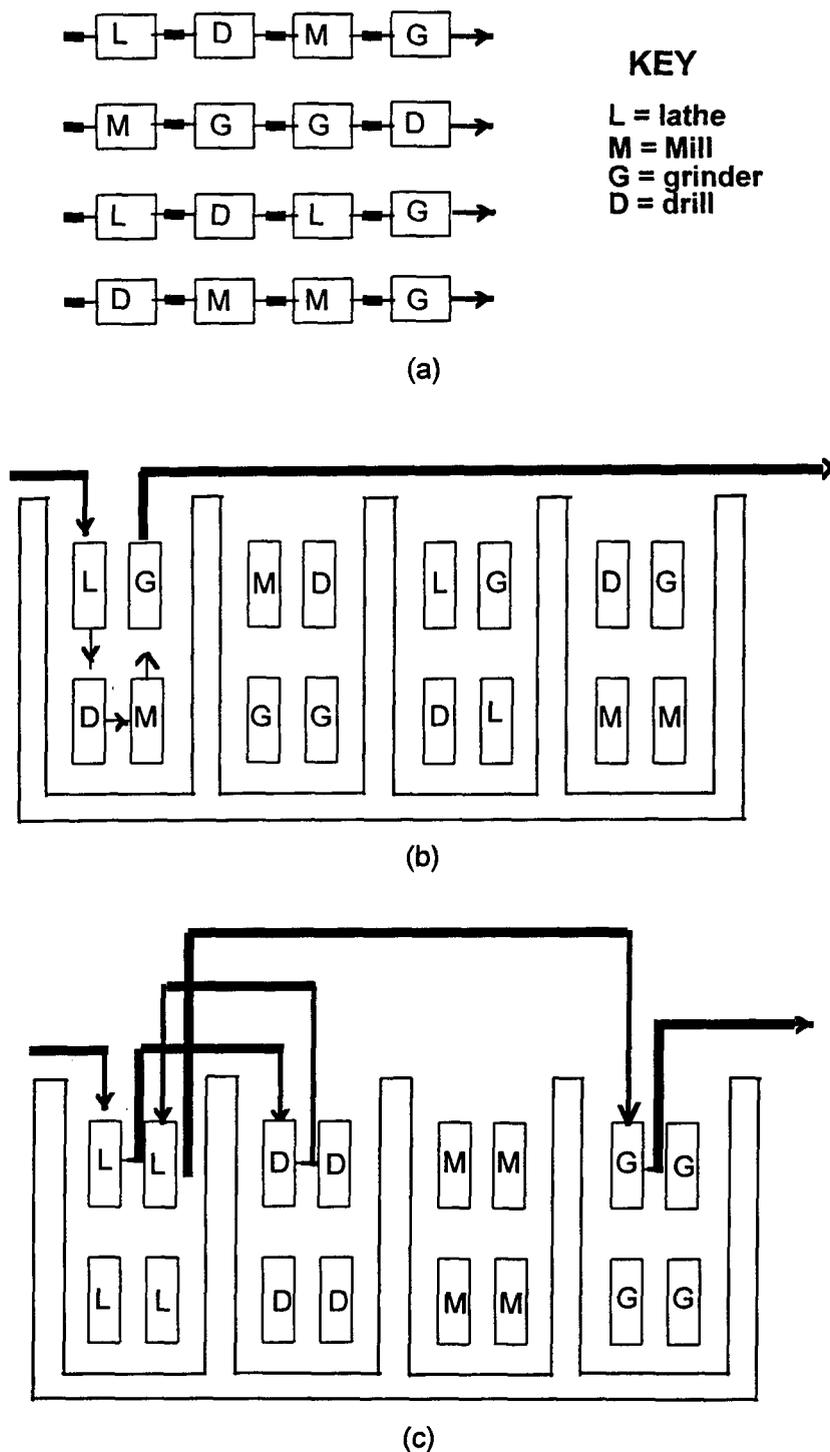


Figura 2.1: Três tipos de arranjo físico: (a) linha; (b) grupo, (c) funcional

Contudo, a transformação do arranjo físico funcional ou de processo em celular implica na duplicação de algumas máquinas que poderão ter capacidade ociosa e/ou poderão ser utilizadas para evitar congestionamento no fluxo de produção (máquinas gargalos). (AL-Qattan, 1990)

Na prática podemos aplicar Tecnologia de Grupo de duas maneiras: a) arranjo físico lógico (também chamado de célula virtual), onde máquinas

são alocadas às famílias sem mudar suas posições; ou b) arranjo físico das máquinas, onde diferentes máquinas são agrupadas para formar células de manufatura de maneira a explorar a eficiência do fluxo. [Kusiak, 1985, 1987].

A Tecnologia de Grupo tem sido adotada em vários países de formas as mais variadas. Ela tem sido utilizada como um componente para a aplicação da Robótica, na Fabricação Assistida por Computador (CAM) e no Projeto Assistido por Computador (CAD). É dentro deste contexto que se pretende apresentar as novas perspectivas de TG como integração entre todos os setores da produção.

2.1 Manufatura Celular

A Tecnologia de Grupo fundamenta-se na idéia de se agrupar atividades repetitivas similares. É então uma idéia universal e um domínio de aplicação ilimitado. A Tecnologia de Grupo pode ser aplicada a uma gama muito ampla das atividades humanas. Abre-se, então, para os meios de produção de bens de consumo, uma nova área de atividade de pesquisa para otimizar seus métodos de manufatura. Uma área em franco desenvolvimento (Devries et al, 1976; Hyer,1984) é a organização da produção em Células Flexíveis de Manufatura (FMC).

A organização da produção em Células de Manufatura consiste em agrupar as peças com características de projeto e produção semelhantes, em famílias de peças conjuntamente com seus planos de execução e as máquinas que as fabriquem. Devido à complexidade desse problema e o elevado número de variáveis a considerar, a organização da produção em FMC tem se tornado num intenso campo de atividade de pesquisa científica e tecnológica. As formas tradicionais da organização da produção em departamentos estanques e produção seriada para a formação de estoques vêm sofrendo, nas últimas décadas, os efeitos de seu envelhecimento. As novas técnicas de produção, que deram um enorme impulso na reorganização industrial do Japão, provocaram no Ocidente, principalmente nos EUA e na Europa, uma necessidade de inovação tanto na área de processos como de métodos. O desenvolvimento na área de informática possibilitou a manipulação de um volume enorme de dados. É bom lembrar que não é importante somente o aspecto de acessar este volume de informações, mas também a forma e a velocidade com que são manipulados. Para a produção de um número elevado de itens sob um número variado de planos e executados em um largo número de máquinas, as possibilidades que se abrem de associá-los é praticamente infinita. Disto se aprende que somente recentemente, com o advento de computadores de elevada capacidade e

velocidade de processamento, tornou-se possível o estudo prático destes problemas. Porém, mesmo o aumento da capacidade e velocidade dos computadores não são ainda suficientes para inovarem os meios da produção. A eletrônica, a micro mecânica e a automação, que tiveram avanço consideráveis neste início de fim de século, completam o quadro, trazendo para o ambiente fabril a automação dos processos. Segundo Mondem (1983) e Schonberg (1982), no Japão para a melhor competitividade industrial, o uso destas novas tecnologias foram utilizadas na implantação do Just-in-Time (JIT), o qual faz uso de Célula de Manufatura. A robótica e outras formas automatizadas nos processos de sistemas de produção cada vez mais irão fazer uso destas Células. A utilização eficiente e eficaz dessa nova filosofia de produção depende da estrutura celular. Examinada por este ângulo não há qualquer dúvida de que a Célula de Manufatura representa uma enorme inovação nas organizações industriais em geral.

A manufatura celular nada mais é do que a divisão física das máquinas da instalação em células de manufatura de produção. Cada célula de produção é responsável por uma família de peças, que por sua vez exige máquinas, operações de máquinas, ferramentas, etc. Uma peça designada para uma célula poderá precisar de todas as máquinas ou uma parte das máquinas de célula.

O ambiente fabril ideal consiste de células independentes, mas uma segregação de máquinas e peças em células sem nenhuma interação entre elas, é muito difícil de se obter devido à ocorrência de peças excepcionais e máquinas gargalos. (Vohra et al, 1990)

Peças excepcionais são aquelas que são manufaturadas em mais de uma célula, e máquinas gargalos são aquelas que existem em mais de uma célula. Burbidge (1975) definiu máquinas gargalos "como aquelas que são altamente sobrecarregadas ou com capacidade disponível próximo ao seu máximo, e neste caso a solução é feita pela adição de uma nova máquina".

Uma grande quantidade de métodos para a formação de células de manufatura foi desenvolvida, e os profissionais que atuam na área costumam classificá-la baseados na maneira como as famílias de peças e grupos de máquinas são identificados (Wemmerlov et al, 1986). Esta prática recomenda a classificação nas seguintes três categorias:

- 1- Identificar grupos de máquinas baseados na similaridade do fluxo da peça na fábrica e então alocar as peças nas máquinas. É conhecida como agrupamento de máquinas.

Pode ser dividida em:

- a) procedimentos não algoritmicais, por exemplo, De Beer et al (1976, 1978). Burbidge (1975) e De Witte, (1979) basearam se na inspeção visual das matrizes construídas;
- b) procedimentos algoritmicais, por exemplo, McAuley (1972) baseia-se em um coeficiente de similaridade que mede a relação entre um par de máquinas.

Esta metodologia para o problema de formação de células é mais efetiva porque o número de peças é usualmente maior do que o número de máquinas.

2 - Identificar família de peças e então alocar máquinas nas famílias de peças. Esta metodologia é conhecida como agrupamento de família de peças.

Pode ser dividida em (Han and Ham, 1987):

- a) sistema de classificação e codificação - aloca um código a uma peça (baseado em curtas características da peça) e peças com códigos semelhantes formam uma família de peças, por exemplo, Choobineh, 1988;
- b) análise de agrupamento - utiliza uma taxonomia numérica para encontrar uma família de peças, por exemplo, Carrie (1973).

3 - identificar uma família de peças e grupo de máquinas simultaneamente. Esta metodologia é conhecida como agrupamento peça - máquina. Por exemplo, Burbidge (1973), King (1980), King and Nakornchai (1982), Chan and Milner (1982), e Chandrasekaran and Rajagopalan (1986 a, b; 1987).

Esta última metodologia é computacionalmente menos atrativa por causa do tamanho excessivo da formulação, mas apresenta a vantagem de produzir uma melhor solução, pois ela procura simultaneamente agrupar peças e máquinas.

2.2 Métodos para a Formação de Famílias de Peças

De acordo com Han et al (1987) existem três métodos para a formação de famílias :

- a) inspeção visual;
- b) método de análise do fluxo de produção;
- c) sistema de classificação e codificação;
- d) métodos de programação matemática (análise de agrupamento, reconhecimento de padrões, conjuntos difusos, redes neuronais, etc...).

Na inspeção visual, a formação de famílias é feita visualmente de acordo com sua aparência. O método é baseado na experiência do operador, o que leva, às vezes, à formação ineficiente de famílias, tornando-se um método subjetivo e inexato. Não é viável para a aplicação em grandes instalações.

A análise do fluxo de produção é um dos métodos mais antigos e utilizados para a formação de família de peças. Este método foi desenvolvido por Burbidge (1973 b), que estava preocupado com a seqüência do processo de produção. A análise do fluxo de produção é uma técnica que utiliza as folhas de processo de componentes que apresentam os mesmos processos de fabricação. A operação contém informações sobre a seqüência da mesma, a designação das máquinas nos postos de trabalho para cada operação. A deficiência do método de análise do fluxo de produção baseia-se no fato de que as folhas do processo são geradas por diferentes pessoas, que têm estilos diversos para o desenvolvimento dos planos de processo. A idéia é formar uma matriz que relacione as operações de cada família com as máquinas necessárias ao seu processamento. Este método torna-se complicado quando se tem um número elevado de componentes e/ou máquinas a serem utilizados.

O sistema de classificação e codificação é o processo no qual as peças não designadas por classes pré-determinadas e a codificação é a atribuição de símbolos para as classes definidas pela classificação. Embora os termos classificação e codificação sejam freqüentemente usados juntos com TG, ambos os termos têm significados distintos.

A classificação é uma técnica usada para organizar uma lista de itens numa ordem lógica e sistemática de classes de acordo com um conjunto específico de critérios de similaridade. A codificação, por sua vez, é a designação arbitrária de um ou mais símbolos para uma peça, os quais, quando interpretados, transmitem informações específicas a respeito da natureza das classes. Tanto a classificação como a codificação não são definidas como uma

filosofia de manufatura, e, portanto, não devem ser confundidas com Tecnologia de Grupo.

O método de classificação e codificação é baseado na classificação das peças em famílias pelo exame dos seus desenhos e/ou atributos de classificação, com o objetivo de agrupar peças semelhantes através de um código. Este código mostra, através de seus dígitos, quais os atributos das peças, facilitando o processamento destas informações. A escolha de um sistema de classificação depende de vários fatores, sendo necessário conhecer em detalhe todas as suas implicações. É um método preciso, porém o sistema selecionado pela companhia deve ir ao encontro das necessidades conjuntas tanto da parte do projeto como da fabricação.

Um sistema de classificação e codificação pode ter outros usos além de desenvolver uma família de peças por Tecnologia de Grupo. É possível, com a utilização de um sistema de classificação e codificação adequado, a redução da duplicação de desenhos, pois a avaliação do código de uma determinada peça que se pretende desenhar permite averiguar se já existe alguma peça igual ou parecida. Também é utilizado para a recuperação de planos de processo previamente desenvolvido, os quais são copiados e modificados para a peça que se pretende desenhar.

A classificação pode ser orientada para o projeto ou para o processo. Na classificação orientada para o projeto, as peças são agrupadas em famílias em função de características do projeto. Na classificação orientada para o processo, as peças são agrupadas em função das semelhanças dos processos tecnológicos. Existem alguns sistemas de classificação que combinam às duas orientações.

Segundo Ham et al (1985), existem cerca de 44 sistemas diferentes, entre eles:

BRISCH(UK)	é um código que possui de 4 a 6 dígitos primários e um grupo de dígitos secundários;
DCLASS(USA)	é um código com 8 dígitos primários e não tem uma estrutura fixa, baseado num software;
MICLASS(TNO/Holanda)	é um sistema que consiste de um código híbrido de 12 dígitos com opção de 18 dígitos adicionais, usado para informações específicas da companhia;

CODE (USA)	é um código híbrido de 8 dígitos, o qual pode ser expandido para 12 dígitos para ser utilizado como atributo, tais como: tratamento térmico, dureza, acabamento final, custo de produção e tempos padrão;
OPITZ(West Germany)	este é um código híbrido de 9 dígitos, o qual contém informações sobre projeto e manufatura e que pode ser estendido pela adição de mais 4 dígitos;
KK3(Japan)	é um código com 21 dígitos;
Togoda(Japan)	é um código com 10 dígitos;
TEKLA(Normay)	é um código com 12 dígitos.

Um sistema de classificação e codificação deverá ser flexível o suficiente para permitir expansões futuras, de maneira a adequá-lo às necessidades específicas da organização. A maioria dos sistemas de classificação e codificação foram desenvolvidos segundo as características de geometria, materiais, dimensões e elementos secundários da peça, mas é necessário esforço para formar uma família de peças. Tais características são então expressas dentro de um código numérico através de um ou mais dígitos.

Normalmente, os sistemas de codificação existentes podem ser classificados em três tipos de estrutura :

- 1 - monocódigos ou hierárquico;
- 2 - policódigos ou atributos;
- 3 - multicódigo ou híbrido.

O monocódigo ou hierárquico é um código hierárquico na estrutura, onde o significado de cada dígito é determinado pela informação dada pelo dígito anterior. Cada caracter amplia a informação transmitida pelo caracter anterior , isto é, cada caracter é dependente da informação do caracter anterior. Um exemplo deste sistema é o utilizado pelas bibliotecas para a catalogação dos livros, o sistema decimal Dewey. Ele é mais utilizado quando temos uma grande diversidade de itens, pois pode prover uma grande quantidade de informações num pequeno número de dígitos.

O policódigo é um código em que cada dígito contém informações próprias e não qualifica a informação dada pelos outros dígitos, ou seja, cada caracter transmite uma informação independentemente de qualquer caracter.

Quando a diversidade de itens é grande, o comprimento pode tornar-se inaceitável.

O multicódigo é um sistema híbrido dos dois primeiros, e é o mais utilizado nos dias atuais.

Com a utilização de um sistema de classificação e codificação eficiente é possível também reduzir a duplicação de desenhos, através da avaliação do código de uma determinada peça, o qual se deseja desenhar, permitindo a averiguação da existência de alguma peça igual ou parecida, ficando as peças do mesmo número de código juntas, dando origem às famílias.

Alguns métodos para a formação de família de peças foram revistos. Os métodos clássicos como os baseados na inspeção visual, na análise de fluxo de produção e nos sistemas de codificação podem ser aplicáveis com ou sem a utilização de procedimentos computacionais, embora para as indústrias modernas existam outros métodos mais apropriados, que serão vistos nos próximos capítulos.

2.3 Métodos de Agrupamento

Atualmente existem numerosos métodos que utilizam a análise de agrupamentos. Ela é utilizada para a formação de famílias de peças e de células de máquina.

Os métodos de agrupamentos podem ser divididos em métodos baseados em arranjo e método análise de agrupamento, dependendo do uso ou não do coeficiente de similaridade.

2.4 Técnicas baseadas em Arranjo

Nas mais diversas técnicas baseadas em Arranjo, a formulação do problema exige a construção de uma matriz que relacione as peças com as máquinas, onde se procura através do rearranjo entre linhas e colunas o agrupamento das peças em famílias.

Este tipo de formulação constrói uma matriz X_{ij} que relaciona as peças às peças (j) com as máquinas (i). Colocando peças em colunas e máquinas em linhas, a matriz será formada "0" e "1", sendo que o "1" representa a visita de uma peça a uma determinada máquina. Este procedimento permite que através da transformação da matriz inicial por permutação de linhas e colunas,

Dentro desta técnica, podemos citar os seguintes algoritmos: o ROC (Rank Order Clustering), o BEA (Bond Energy Analysis) e o DCA (Direct Clustering Analysis).

2.4.1 O método BEA (Bond Energy Analysis)

BEA é um algoritmo geral de agrupamento desenvolvido por Mc Cormick et al (1972), o qual foi utilizado para uma variedade de aplicações. Esta metodologia gera grupos diagonalizados baseado no produto dos elementos adjacentes na matriz máquina - peça, chamada energia de ligações. (bond energy)

Se entre cada par de linhas e colunas vizinhas existem células positivas na matriz, dizemos que existe uma afinidade e cria-se uma energia de união, a qual é produto dos valores dos pares dos elementos adjacentes, como mostrado na alternativa abaixo:

a) elementos de linha adjacentes, X_{ij} , $X_{i,j+1}$

Alternativas	X_{ij}	$X_{i,j+1}$	Bond Energy
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

b) elementos de coluna adjacentes X_{ij} , $X_{i-1,j}$

Alternativas	X_{ij}	$X_{i-1,j}$	Bond Energy
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

A metodologia inicia selecionando-se uma coluna ou linha, arbitrariamente, razão pela qual podemos ter uma infinidade de soluções.

Num primeiro passo, esta metodologia procura rearranjar as colunas. A coluna (ou linha) com grande contribuição para uma energia de

ligação total são permutadas de maneira a direcionar os elementos de maior afinidade juntos.

Veja abaixo, na figura 2.4, a matriz do problema original e da solução final após a aplicação do método (King, 1980).

		P	E	Ç	A			P	E	Ç	A	7				
		1	2	3	4	5	6	RBE								
M	1			1		1			M	5	1	1	1	2		
Á	2		1	1				1	Á	3	1	1		1		
Q	3	1			1				Q	4			1	1	1	2
U	4		1	1		1		1	U	2			1	1		1
I	5	1			1		1		I	1			1		1	
N									N							
A									A							
	CBE		1				3		CBE	1	1		2	1		11

Figura 2.4: Método BEA - rearranjo de matrizes

É utilizada uma medida de efetividade (ME) no qual quanto maior este índice, maior a densidade dos elementos.

A expressão abaixo calcula o ME para um arranjo $A(m,n)$ mostrando o vínculo deste que existe com seus vizinhos na corrente disposição.

$$ME(A) = 1/2 \left[\sum_{i=1}^m \sum_{j=1}^n (a_{i,j} + a_{i,j+1} + a_{i,j-1} + a_{i+1,j} + a_{i-1,j}) \right] \quad (2.1)$$

onde: A é qualquer arranjo $M \times N$ não negativo;

$$A_{0,j} = A_{M+1,j} = A_{i,0} = A_{i,N+1} = 0, \text{ isto é, os vizinhos externos são nulos.}$$

Elementos excepcionais são elementos pertencentes a um grupo, mas requerendo processamento em máquinas de outros grupos (King, 1980).

2.4.2 O método de Agrupamento ROC (Rank Order Clustering)

É uma metodologia iterativa desenvolvida por King (1980), que gera um bloco diagonal na matriz máquina - peça repetidamente rearranjando as colunas e linhas da matriz máquina/peça de acordo com valores binários. Esta metodologia apresenta depois de um número finito de iterações uma matriz onde as linhas e colunas são dispostas em ordem decrescente de valor quando lidos como palavras binárias (Chu, 1989).

O método é ilustrado abaixo (figura 2.5) com aplicação de um exemplo de King(1980),e após uma série de iterações temos a matriz resultante máquina-peça.

PESOS BINÁRIOS } M	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	DEC. EQUIV.	RANK
	P	E	Ç	A	S			
Á	1	2	3	4	5	6		
Q	1		1		1		10	5
U	2	1	1				24	4
I	3	1		1			36	2
N	4	1	1		1		26	3
A	5	1		1		1	37	1
S								

Figura 2.5: Método ROC - matriz inicial

PESOS BIN. ⌋		M		P	E	Ç	A	S
		Á	1	2	3	4	5	6
2 ⁴		Q	5	1		1		1
2 ³		U	3	1		1		
2 ²		I	4		1	1	1	
2 ¹		N	2		1	1		
2 ⁰		A	1			1	1	
	S							
	DECIMAL EQUIV. } S		24	6	7	24	5	16
	RANK		1	5	4	2	6	3

Figura 2.6: Método ROC - Matriz intermediária

	1	4	6	3	2	5	DEC / EQUIV.	RANK
5	1	1	1				56	1
3	1	1					48	2
4				1	1	1	7	3
2				1	1		6	4
1				1		1	5	5
DEC. EQUIV. } RANK	24	24	16	7	6	5		
	1	2	3	4	5	6		

Figura 2.7 : Método ROC - matriz final

Embora o algoritmo seja simples e fácil de ser implementado no computador apresenta as seguintes desvantagens: 1) o resultado final depende da disposição da matriz inicial; 2) o valor binário, que é usado para o rearranjo restringe o tamanho do problema que pode ser tratado pela técnica; 3) não pode fornecer finalmente grupos mutuamente independentes e; 4) uma inabilidade para analisar matrizes grandes, pois o comprimento da palavra binária aumenta.

Este algoritmo pode tratar os problemas das máquinas gargalos e/ou elementos excepcionais.

Uma versão revisada do ROC chamada ROC 2, foi desenvolvida por King and Nakornchai (1982) de maneira a aumentar a eficiência computacional e superar as limitações.

2.4.3 O Algoritmo de Agrupamento Direto (DCA)

Esta metodologia foi desenvolvida por Chan e Milner (1982), e procura agrupar famílias pela formação de blocos através do rearranjo das linhas e colunas. O primeiro passo consiste na contagem do número de células positivas, isto é, com entrada 1 em cada linha e coluna e rearranjo, a matriz com as linhas em ordem crescente de K e as colunas em ordem decrescente de K.

As linhas com as células positivas "mais à esquerda" são colocadas no topo e as colunas com as células positivas "mais a cima" são movimentadas para a esquerda da matriz. Após uma série de iterações e depois que nenhuma modificação na matriz é detectada, é formado um bloco diagonal desde a extremidade esquerda superior até a extremidade direita inferior. (Chu et al, 1990).

Este algoritmo procura solucionar o problema das máquinas gargalos e/ou elementos excepcionais. DCA é muito semelhante ao ROC na aplicação de agrupamento máquina-peça.

2.5 Método de Análise de Agrupamento (Cluster analysis-based methods).

Basicamente o objetivo do presente método é a divisão de uma coleção de objetos em grupos baseados em alguns critérios ou medidas de similaridade, onde um objeto é representado por certos números de características (Burbidge, 1975; Chu, 1989; Seifoddini, 1989).

Dado o vetor padrão de n objetos X_1, X_2, \dots, X_n num espaço R^n , o processo de agrupamento procura dividir o espaço padrão em subregiões $R^{n_1}, R^{n_2}, \dots, R^{n_m}$ de maneira que cada objeto caia dentro de uma e somente uma destas regiões.

Isto é, sujeito às seguintes restrições:

$$R^{n_1} \cup R^{n_2} \cup \dots \cup R^{n_m} = R^n \quad (2.2)$$

$$R^{n_i} \cup R^{n_j} \neq 0, \quad i \neq j \quad (2.3)$$

Em estatística, um padrão (uma peça neste contexto) é expressada usualmente como um vetor X , $X_1 = [x_1, x_2, \dots, x_n]^T$, onde cada elemento de vetor representa um atributo dimensional. Neste contexto, X é uma peça, e $x_i = 1, 2, \dots, n$, são planos de processo associadas com máquina, ou seja, se uma máquina é requerida para um dado processo, $x_i = 1$, ao contrário, $x_i = 0$.

Freqüentemente, as metodologias usam como medida da similaridade padrão uma função distância de uma função similaridade. Os coeficientes de similaridade dão maior flexibilidade ao processo de agrupamento máquinas-peças, e facilitam a aplicação de recursos computacionais.

2.5.1 - Função Distância

A similaridade ou a diferença entre duas peças pode ser definida em termos de distância entre elas. A distância entre dois vetores padrões X_i e X_j chamada, d_{ij} , é função real simétrica, que obedece aos seguintes axiomas:

$$\text{- Reflexividade } d_{ij} = 0 \quad (2.4)$$

$$\text{- Simétrica } d_{ij} = d_{ji} \quad (2.5)$$

$$\text{- Inequalidade triangular } d_{ij} \leq d_{is} + d_{sj} \quad (2.6)$$

onde s é qualquer peça diferente de "i" e "j".

Existem várias maneiras para definir a função d_{ij} , entre elas temos: a distância métrica de Minkowski, a distância ponderada de Minkowski e a distância de Hamming (Gu e ElMaraghy, 1989):

I - Distância métrica de Minkowski

$$d_{ij} = |X_i - X_j| = \left[\sum_{k=1}^n |X_{ik} - X_{jk}|^r \right]^{1/r} \quad (2.7)$$

onde: X_{jk} = é o valor da variável késsimo para o caso i
 n = o espaço dimensional
 r = um inteiro positivo

Dois casos especiais são largamente usados,

$r = 1$, distância absoluta
 $r = 2$, distância Euclidiana

II - Distância ponderada de Minkowski

$$d_{ij} = \left[\sum_{k=1}^n W_k |X_{ik} - X_{jk}|^r \right]^{1/r} \quad (2.8)$$

onde W_k = peso

III - Distância de Hamming

$$d_{ij} = \sum_{k=1}^n \delta(X_{ik}, X_{jk}) \quad (2.9)$$

$$\text{onde: } \delta(X_{ik}, X_{jk}) = \begin{cases} 1, & \text{se } X_{ik} \neq X_{jk} \\ 0, & \text{ao contrário} \end{cases} \quad (2.10)$$

Isto é, podemos concluir que dada duas peças i e j e seus respectivos vetores, a função distância de Hamming é definida por:

$$d_{ij} = \sum_{k=1}^n |X_{ik} - X_{jk}| \quad (2.11)$$

Concluimos que quanto menor d_{ij} , maior a similaridade entre X_i e X_j . Outro método utilizado é a soma dos erros quadrados (Gu e ElMaraghy, 1989):

$$J = \sum_{j=1}^{N_c} \sum_{x \in S_j} \|x - \mu_j\|^2 \quad (2.12)$$

onde: N_c = é o número do domínio do agrupamento, isto é, de família de componentes;

S_j = é o conjunto de amostra pertencente ao domínio j-ésimo, ou seja, o j-ésimo grupo de componentes nesta pesquisa;

μ_j = é a média do vetor componente do conjunto S_j ;

O valor de μ_j é determinado como:

$$\mu_j = 1/N_j \sum_{x \in S_j} x \quad (2.13)$$

2.5.2 Função Similaridade

Baseado na medida de similaridade entre os padrões representados no plano de processo, um critério de agrupamento é requerido. Este critério de agrupamento usado é baseado na otimização de um índice de desempenho, o qual divide os dados fornecidos em grupo de máquinas e famílias de peças.

Um exemplo de funções de similaridade é o coeficiente de Tanimoto, (Bow 1984), o qual é dado por:

$$S(X, Z) = X^T Z / X^T X + Z^T Z + X^T Z \quad (2.14)$$

onde :

X^T é a transposta do vetor X

Se X e Z são vetores binários, a expressão acima mostra a razão entre o número de características comuns entre X e Z para o número de características que pode ser qualquer um dos vetores binários X, Z .

Outra medida de similaridade foi desenvolvida por Vila Filho (1987), onde dado duas peças: X e Y , o coeficiente de similaridade $S(x,y)$ é expresso:

$$S(x,y) = n(x - y) / N(x \cap y) \quad (2.15)$$

onde:

$n(x-y)$ = número de máquinas requeridas pelo componente X e não requeridas pelo componente Y;
 $N(x \cap y)$ = número de máquinas requeridas por ambos componentes.

Para ilustrar temos dois componentes X e Y, representados por suas respectivas máquinas, sendo que $X = (A,B,E,F,G)$ e $Y = (A,B,C,D,E)$,

onde

$$\begin{aligned} n(X - Y) &= (F,G) \\ N(X \cap Y) &= (A,B,E) \end{aligned}$$

logo: $S(X,Y) = 2/3$

Podemos concluir que se:

$S(X, Y) = 0$; o lote de máquina do componente X está contido no lote de máquinas do componente Y.

$n(X \cap Y) = 0$; os componentes X e Y deverão ser fabricados em células distintas.

As análises de agrupamento podem ser também caracterizadas como métodos aglomerativos ou divisivos, conforme a seqüência de agrupamento.

O método aglomerativo começa por um padrão isolado e aglutina padrões parecidos de acordo com um valor limite (threshold) de baixo para cima, de maneira a formar hierarquias. Entretanto, um algoritmo divisivo inicia pelo conjunto de todos os padrões e divide ao longo da direção de cima para baixo em um número de agrupamento de acordo com um critério pré-estabelecido.

Mc Auley (1972) foi o primeiro a introduzir SLCA (Single Linkage Cluster Analysis) para o problema de formação de famílias de peças em análise de fluxo de produção. Este algoritmo, SLCA (Single Linkage Cluster Analysis) foi desenvolvido por Sneath (1973), e utiliza um coeficiente de similaridade para agrupar peças que requerem um processo semelhante.

Basicamente a metodologia adotada tem como objetivo calcular para cada par de máquinas um coeficiente de similaridade, o qual procura descrever a semelhança existente entre duas máquinas, em termos do número de peças, que são processadas por ambas as máquinas, e o número de peças visitando cada máquina.

Este coeficiente pode ser expresso por

$$S_{jk} = N_{Jk} / N_{Jk} + U \quad (2.16)$$

onde:

$$U = N_{Jk} + N_{jk} \quad (2.17)$$

N_{Jk} = número de peças que visitam ambas as máquinas;

N_{jk} = número de peças que visitam uma ou outra das máquinas, mas não ambas;

S_{jk} = coeficiente de similaridade entre a máquina j e k.

Como ilustração, consideramos a seguinte matriz máquina - peça:

	P	E	Ç	A	S	
M						
Á	1	2	3	4	5	6
Q	1	1	1		1	1
U	2	1		1		
I	3	1	1		1	
N	4	1			1	1
A	5			1		
S						

Figura 2.8: Matriz inicial máquina-peça

O coeficiente de similaridade para todos os pares de máquinas são dados por:

$$S_{1,2} = 1/1+4 = 1/5 = 0,20$$

$$S_{1,3} = 3/3+1 = 3/4 = 0,75$$

$$S_{1,4} = 2/2+3 = 2/5 = 0,4$$

$$S_{1,5} = 1/1+3 = 1/4 = 0,25$$

$$S_{2,3} = 1/1+4 = 1/5 = 0,2$$

$$S_{2,4} = 2/2+3 = 2/5 = 0,4$$

$$S_{2,5} = 2/2+1 = 2/3 = 0,67$$

$$S_{3,4} = 2/2+3 = 2/5 = 0,4$$

$$S_{3,5} = 1/1+3 = 1/4 = 0,25$$

$$S_{4,5} = 1/1+4 = 1/5 = 0,2$$

Estes coeficientes são colocados numa matriz de similaridade como a apresentada na figura 2.9.

	M	Á	Q	U	I	N	A	S
M			1	2	3	4	5	
Á	1		--	0,20	0,75	0,40	0,25	
Q	2			--	0,20	0,40	0,67	
U	3				--	0,40	0,25	
I	4					--	0,20	
N	5						--	
A								--
S								

Figura 2.9: Matriz similaridade para a figura 2.8

As duas máquinas que formam o agrupamento inicial são tiradas da matriz de similaridade, o qual apresenta o maior coeficiente de similaridade. A matriz de similaridade é então examinada novamente para localizar o maior coeficiente restante. As máquinas associadas são agrupadas juntas. A metodologia é repetida até que todas as máquinas associadas sejam agrupadas juntas em um único grupo, ou até o momento, onde os coeficientes de similaridade estão abaixo de um nível especificado (threshold).

O processo de agrupamento pode ser graficamente visualizado usando um dendograma que proporciona uma maneira mais descritiva de mostrar resultados. Dendograma é um diagrama de árvores cujos ramos

representam máquinas, e as linhas horizontais conectadas aos ramos representam o valor limite, o qual o grupo de máquinas é formado.

O dendograma para a figura 2.9 é apresentado na figura 2.10, supondo que sejam aceitos coeficientes de similaridade para valores até 0,4.

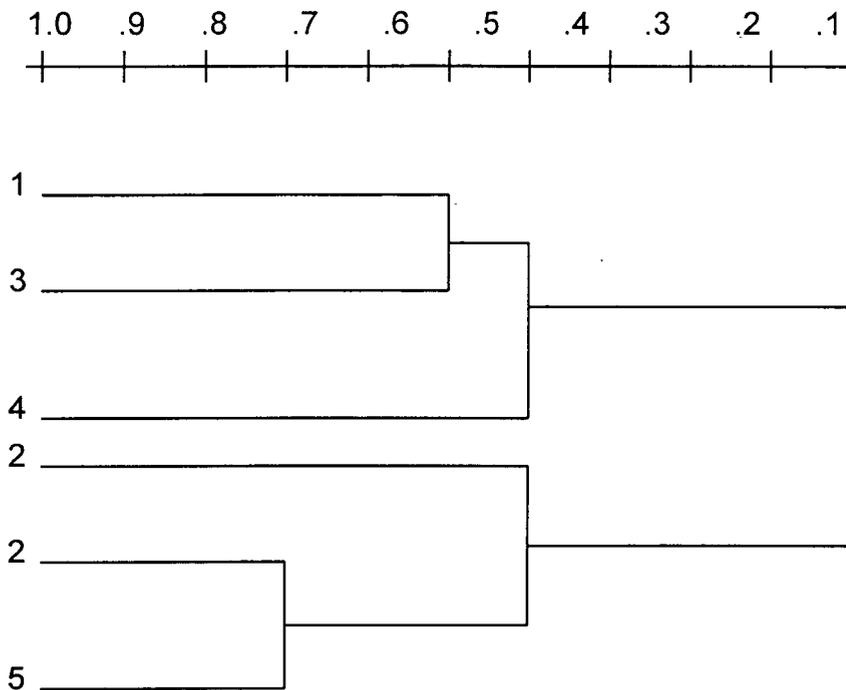


Figura 2.10: Dendograma para os valores da figura 2.9

Então, para um valor limite de 0,4, por exemplo, foram definidos quatro agrupamentos de máquinas que são: (1,3,4) e (2,4,5). A decisão de onde cortar o dendograma é usualmente deixada a critério do utilitário.

Embora o algoritmo apresente uma série de vantagens, entre elas a de proporcionar um agrupamento de máquinas de maneira estrutural (o qual é significativo quando temos o envolvimento de muitas máquinas) e facilidade computacional, apresenta algumas desvantagens, entre elas temos o problema de encadeamento, o qual permite que dois ou mais agrupamentos possam estar unidos porque existem similaridade entre dois de seus membros, embora a maioria deles tenha baixos coeficientes de similaridade. Outra desvantagem é a necessidade de maiores informações de maneira a determinar quando parar o processo de formação de grupos.

Vários trabalhos foram desenvolvidos, podendo citar o desenvolvido por Anderberg (1973), o qual é adicionado pesos as variáveis. Seu coeficiente de similaridade é definido por

$$S_{ij} = 2J_k / 2J_k + U \quad (2.18)$$

O problema de encadeamento descrito acima poderá ser reduzido pela utilização do método ALCA (Average Linkage Clustering). Este método elimina a possibilidade de uma máquina ser alocada a um grupo simplesmente devido a um alto nível de similaridade com uma das máquinas do grupo. Uma das muitas maneiras de calcular o coeficiente médio de similaridade foi desenvolvida por Sokalfand Michner (1973), e é definido como:

$$S_{ij} = s_{ij} / N_i \times N_j \quad (2.19)$$

S_{ij} = coeficiente de similaridade entre dois grupos "i" e "j";

s_{ij} = soma de coeficientes de similaridade aos pares entre todos os membros de dois grupos;

N_i, N_j = o número de máquinas no grupo i e j, respectivamente.

Por exemplo, se o grupo i é composto pelas máquinas A, B e C, e o grupo j pelas máquinas D e E, então o coeficiente de similaridade é dado por:

$$S_{ABC,DE} = S_{A,D} + S_{A,E} + S_{B,D} + S_{B,E} + S_{C,D} + S_{C,E} / N_{ABC} \times N_{DE} \quad (2.20)$$

Como vimos, um grande número de coeficientes de similaridade e algoritmos de agrupamentos são disponíveis, e a escolha do algoritmo depende do tipo de problema.

Como conclusão o processo de formação de grupo de máquina poderá ser usado tanto pela análise da matriz peça-máquina como pelo método do coeficiente de similaridade. O método de análise peça máquina para a solução de pequenos problemas gera grupos de peça-máquina mais rapidamente e eficientemente. No caso de resolução de grandes problemas, o método do coeficiente de similaridade é mais efetivo, pois incorpora dados de manufatura e problemas de máquinas gargalos presentes na linha de produção.

Todos os estudos adotaram o algoritmo de agrupamento hierárquico e o coeficiente de similaridade, e uma das maiores desvantagens destes métodos é que em qualquer estágio do agrupamento quando dois objetos estão juntos no grupo, eles permanecem permanentemente atados mesmo se conduza a um agrupamento não natural. Foi desenvolvido por Chandrasekharan, M.P. et al (1986) um algoritmo não-hierárquico que não apresenta este problema, isto é, ele deixa que o agrupamento natural apareça a partir dos dados sem uní-los a quaisquer dados de estágios anteriores. O objetivo deste método é desenvolver um algoritmo não heurístico de maneira a obter um agrupamento de máquina auto-suficiente, de modo que cada grupo

processe quase todas as peças de uma dada matriz componente. A formação de um grupo de máquinas e família de peças é tratada como um problema de bloco diagonalização de uma matriz peça-máquina. O K-Means e a função "city block" são usadas para medir a dissimilaridade ou similaridade. A metodologia é aplicada em três estágios: num primeiro passo é formulado um gráfico bi-partido e os resultados desta conectividade são usados como o ponto inicial para um agrupamento não hierárquico; no segundo estágio as famílias de peças e os grupos de máquinas são diagonalmente correlatados pelo uso da função objetivo; e o terceiro estágio identifica a "semente ideal" para o agrupamento pelo aperfeiçoamento do critério quantitativo inicial de agrupamento.

2.6 Técnica de Reconhecimento de Padrão

Numa primeira abordagem, o Reconhecimento de Padrão está preocupado com a descrição e análise das medidas feitas do mundo físico ou processo medital (Bow, 1984).

Um dos objetivos comuns de um sistema de reconhecimento de padrões é o reconhecimento de membros de uma dada classe em um conjunto contendo muitas classes de padrões. Um padrão descreve qualquer membro de uma categoria representando uma classe padrão.

Segundo FU(1982), os problemas de Reconhecimento de Padrões podem ser agrupados no enfoque baseado na teoria de decisão e no enfoque sintático.

No agrupamento pela teoria de decisão, um conjunto de medidas características é extraído dos padrões e a alocação (classificação) a uma determinada classe é feita pela divisão do espaço de características, isto é, o conjunto das particularidades que identifica uma classe específica de padrões. Como exemplo desta abordagem, podemos incluir os diagnósticos médicos, classificação de superfícies metálicas, etc...

No caso de que o processo de reconhecimento tenha que incluir não só a capacidade de alocar o padrão a uma classe particular (classificá-lo) como também a capacidade de descrever aspectos do padrão, que tornam inadequado para alocá-lo a alguma outra classe, é proposto a utilização de abordagem sintática. A informação estruturada (hierárquica) é apresentada de forma análoga à sintaxe das linguagens através de uma analogia entre as estruturas dos padrões e a estrutura sintática de uma linguagem (gramática).

A abordagem define primitivas de cada padrão e descreve por meio de um conjunto de regras sintáticas suas relações dentro do padrão em estudo. O

processo de reconhecimento é feito pela efetivação de uma análise sintática (parsing) da sentença que descreve um dado padrão.

A tarefa de reconhecimento de padrões pode ser dividida em três fases: aquisição de dados, pré-processamento de dados e decisão de classificação.

Os dados podem ser representados por um vetor $X(i)$ cujos componentes são as características (features), por exemplo, nas peças podem ter características geométricas e tecnológicas, através de uma matriz binária, assinalando com o valor 1, a ocorrência da característica. Cada elemento do vetor padrão X pode ser descrito como uma variável no espaço n -dimensional, o espaço padrão, e como consequência cada vetor aparece como um ponto no espaço. Para efeito de classificação e agrupamento destes vetores podem ser atribuídos pesos diferenciados em algumas das posições de maneira a destacar algumas das mais significativas características do ponto de vista de complexidade ou interesse para fins de agrupamento (clustering) (Vila Filho, 1987).

Logo, o agrupamento ocorre quando existem padrões com características comuns ou pertencentes a mesma classe. Como consequência, as superfícies que separam estes planos são chamadas hiperplanos, os quais são definidos pelas funções de decisão ou discriminante.

Considere o problema de classificação de padrões mostrado na figura 2.11 onde padrões da classe w_1 e da classe w_2 são separados por linha reta, com equação do tipo:

$$g(x) = w_0 + w_1 x_1 + w_2 x_2 \quad (2.21)$$

onde: $w_0, w_1, w_2 =$ parâmetros ou pesos
 $x_1, x_2 =$ são vetores de entrada ou valores das características

A função saída pode ser definida como:

$$O(x) = \begin{cases} 1, & \text{se } g(x) > 0 \\ 0, & \text{se } g(x) < 0 \end{cases} \quad (2.22)$$

A localização da linha é completamente determinada pelos pesos w_0, w_1, w_2 . O valor absoluto de $g(x)$ nos diz como que um dado vetor entrada encontra-se na superfície de decisão. Isto nos dá um caminho para caracterizar como um conjunto de pesos é bom (Vila Filho, 1987).

Logo, $g(x)$ pode ser usado como uma função de decisão, pois um padrão desconhecido será classificado em w_1 se $g(x) > 0$ e w_2 se $g(x) < 0$.

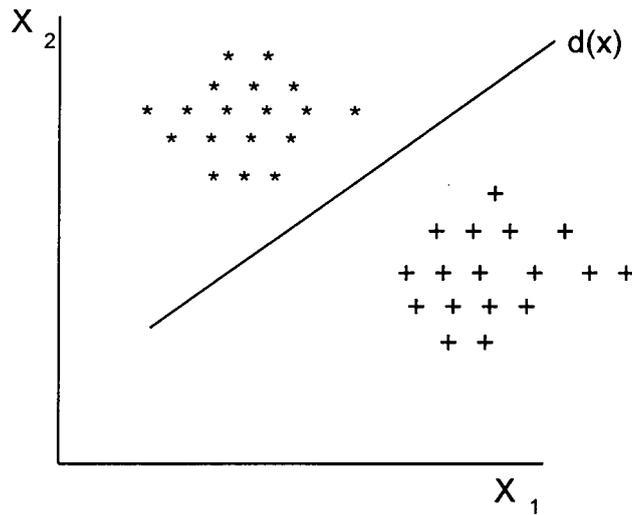


Figura 2.11: Classes de padrões num espaço bidimensional

Peklenik and Grum (1980, 1982, 1984) aplica um cartão codificado binariamente e o método da função potencial para comparar as características da peça classificada com aquelas da peça referência, e automaticamente determinar a família que a peça pertence. A figura 2.12 mostra o procedimento computacional usado por Peklenik e Grum (1980).

Algumas perguntas ainda não são respondidas, tais como decidir a peça referência, como selecionar as características da peça e como determinar o centróide de uma classe. A desvantagem do sistema é a incapacidade de criar uma nova família de peças, quando a peça não se encaixa em alguma das famílias existentes.

Logar e Peklenick (1987) desenvolveram uma metodologia heurística para identificar a peça referência. O algoritmo identifica inicialmente a peça mais complexa no conjunto de dados como a representativa da primeira família de peça, e atribui a esta, a peça menos complexa em termos de característica, possuindo a mesma um conjunto de características pertencentes ao conjunto da peça mais complexa.

A metodologia continua identificando a segunda peça mais complexa entre as restantes e o processo de alocação é repetido formando a segunda peça família. O processo é repetido exaustivamente até que todos os dados estejam alocados. Se o número de família de peças é muito grande, então o método SLCA é aplicado de maneira a tornar as famílias de peças as mais familiares possíveis.

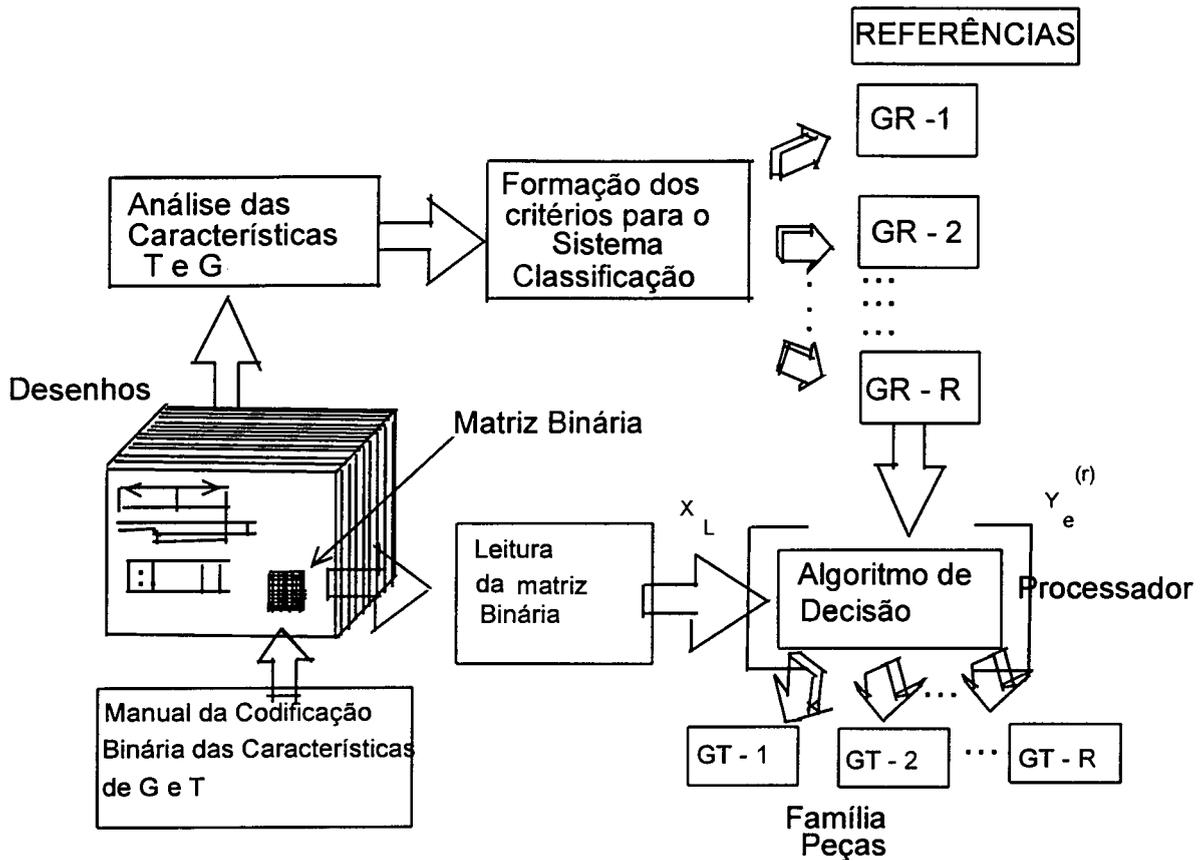


Figura 2.12: Procedimento para classificação auxiliada por computador (Peklenik e Grum, 1980)

Como desvantagem o método pode formar somente uma família de peça. Se por acaso uma peça muito complexa aparecer, Jakubowisk e Kasprzak (1977) desenvolveram uma metodologia para descrever e reconhecer peças rotacionais. Cada peça é descrita por uma combinação de segmentos S_i (primitivas) tiradas de um conjunto de segmentos S , conforme mostrado na Figura 2.13.

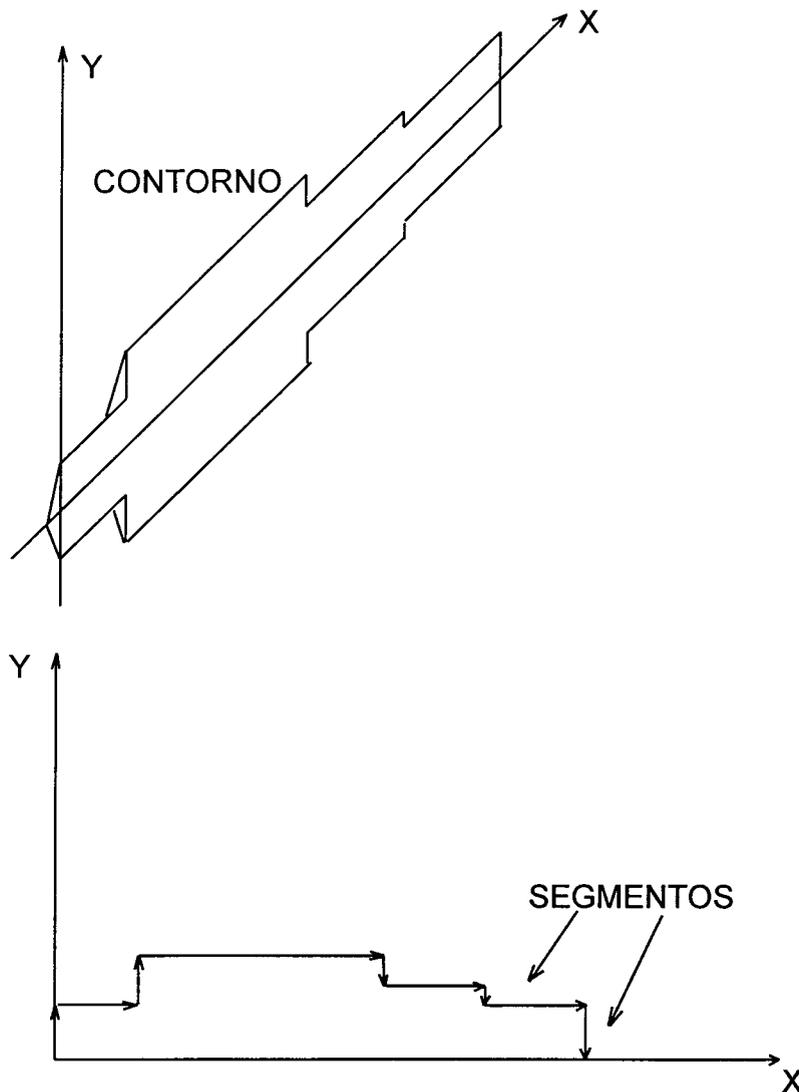


Figura 2.13: Construção de um contorno (Srinivasan et al, 1985).

Eles são representados, basicamente, por diferentes formas de linha e segmentos de arcos, com um ponto inicial, um ponto final, e um ponto central (zero para segmento de linhas) e uma direção. A geometria de uma peça rotacional pode ser definida dando somente a metade superior de um perfil de informação 2D, conforme mostrado na figura 2.14. A representação do contorno da peça é feita na forma de um string padrão. O método é restrito para peças bi-dimensionais (rotacionais), e quando aparecem peças rotacionais com furos axiais e radiais elas não podem ser descritas.

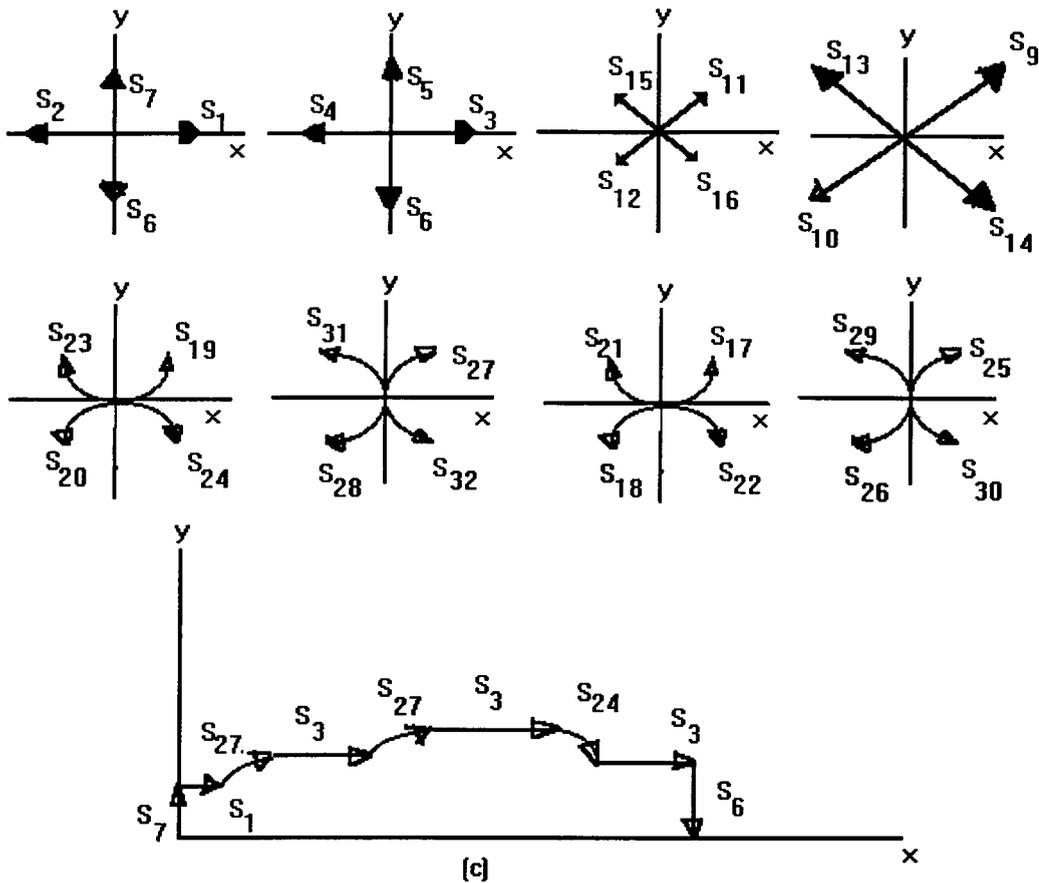


Figura 2.14: Conjunto de segmentos e um contorno (Srinivasan et al, 1985)

Srinivasan et al (1985) desenvolveram uma solução para a formação e classificação de família de peças pela aplicação de técnicas de análise e inferência sintaxe da linguagem formal $L(G)$.

A gramática formal gera strings (sentenças) e parse (análise sintaxe), e o algoritmo determina se uma string particular pertence a uma linguagem ou pode ser gerada por uma gramática. A análise sintaxe é feita utilizando um algoritmo (Earley).

O processo de reconhecimento de uma desconhecida sentença (string) padrão pode ser representado por:

$$x \in L [G(i)], \text{ for } i=1, \dots, m$$

onde:

$G(i)$ = a gramática representativa da linguagem i ;

x = a desconhecida string padrão.

Uma gramática de contexto livre, linear e fase estruturada tem sido escolhida como apropriada para modelar os componentes processados (Fu, 1982):

V_n = conjunto de símbolos não terminais;

V_t = conjunto de símbolos terminais;

P = conjunto finito de regras de produção indicada $a \rightarrow b$, onde "a" e "b" são strings através $(V_n \cup V_t)$ e com o envolvimento de pelo menos um símbolo de V_n ;

$S \subseteq V_n$ = símbolo inicial de uma sentença.

Quatro gramáticas foram construídas para representar as quatro famílias. As primitivas usadas para representar uma matriz máquina-peça estão mostradas na figura 2.15 e um exemplo é dado na figura 2.16 mostrando como verificar se uma peça pertence a uma família de peças.

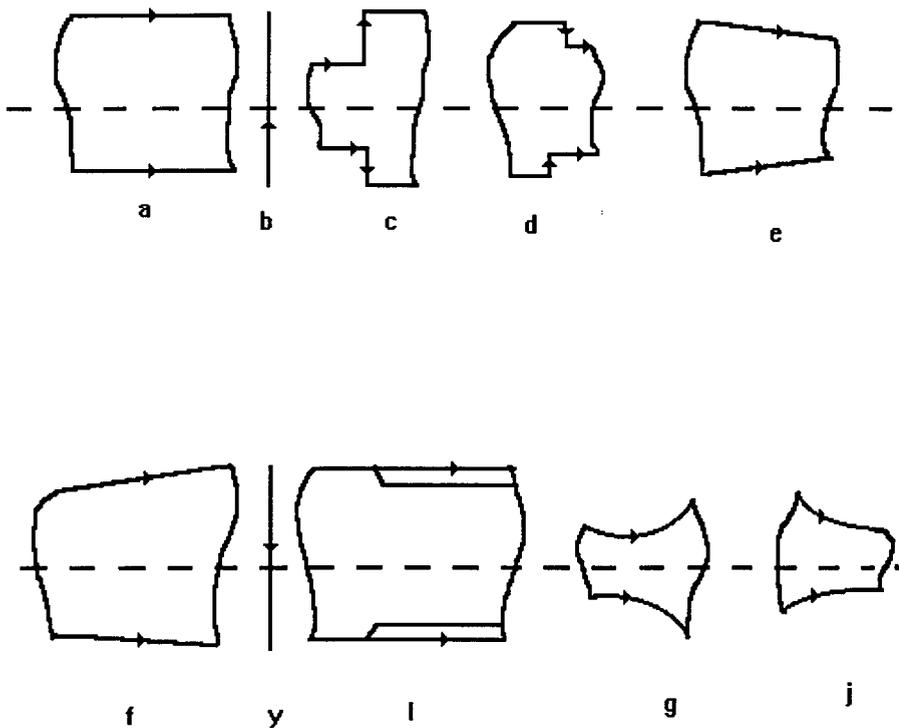
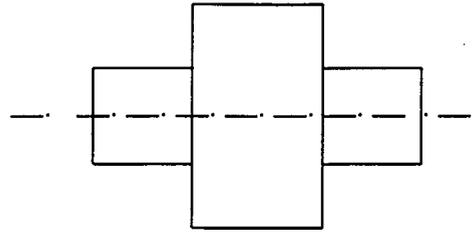


Figura 2.15: Diagrama das primitivas



$S \rightarrow T \rightarrow BGU \rightarrow BXZKU \rightarrow PXZKU \rightarrow bXZKU \rightarrow$
 $bCZKU \rightarrow bcZKU \rightarrow bcQKU \rightarrow bcAKU \rightarrow bcaKU \rightarrow$
 $bcaKy \rightarrow bcaQy \rightarrow bcaDy \rightarrow bcady.$

Figura 2.16: Exemplo de padrão de reconhecimento (Srinivasan et al, 1985).

Como exemplo, a gramática representando a família 1 é dada por:

$$G_1 = (V_n, V_t, P, S)$$

$$V_n = (T, B, G, U, X, Y, Z, K, F, C, A, E, Q, D, P, L, M)$$

$$V_t = (a, b, c, d, e, f, g, j, y)$$

	$S \rightarrow T$	$T \rightarrow BGU$	$G \rightarrow XYZK$	$G \rightarrow XZK$	$K \rightarrow G$
	$X \rightarrow F$	$X \rightarrow C$	$Y \rightarrow F$	$Y \rightarrow A$	$Y \rightarrow E$
	$Z \rightarrow E$	$Z \rightarrow Q$	$Q \rightarrow DA$	$Q \rightarrow DQ \rightarrow A$	
P	$Q \rightarrow DE$	$B \rightarrow PA$	$B \rightarrow P$	$B \rightarrow PF$	$P \rightarrow bP$
	$A \rightarrow AY$	$A \rightarrow aA$	$A \rightarrow L$	$U \rightarrow AY$	$U \rightarrow Y$
	$Y \rightarrow yY$	$F \rightarrow FY$	$F \rightarrow L$	$F \rightarrow fF$	$F \rightarrow f$
	$C \rightarrow cC$	$E \rightarrow eE$	$D \rightarrow dD$	$D \rightarrow M$	$D \rightarrow d$
	$L \rightarrow g$	$M \rightarrow j$	$P \rightarrow b$	$A \rightarrow a$	$Y \rightarrow y$
	$C \rightarrow c$	$E \rightarrow e$			

Neste artigo foi proposto uma extensão ao modelo, com a introdução de uma estrutura de árvore como representação mais completa de superfícies incluindo furos, rasgos, etc.... Na estrutura da árvore, cada superfície, que é residente em uma superfície particular, é representada como uma criança da superfície residente como visto na figura 2.17. O modelo necessita da definição, a priori, das famílias de peças.

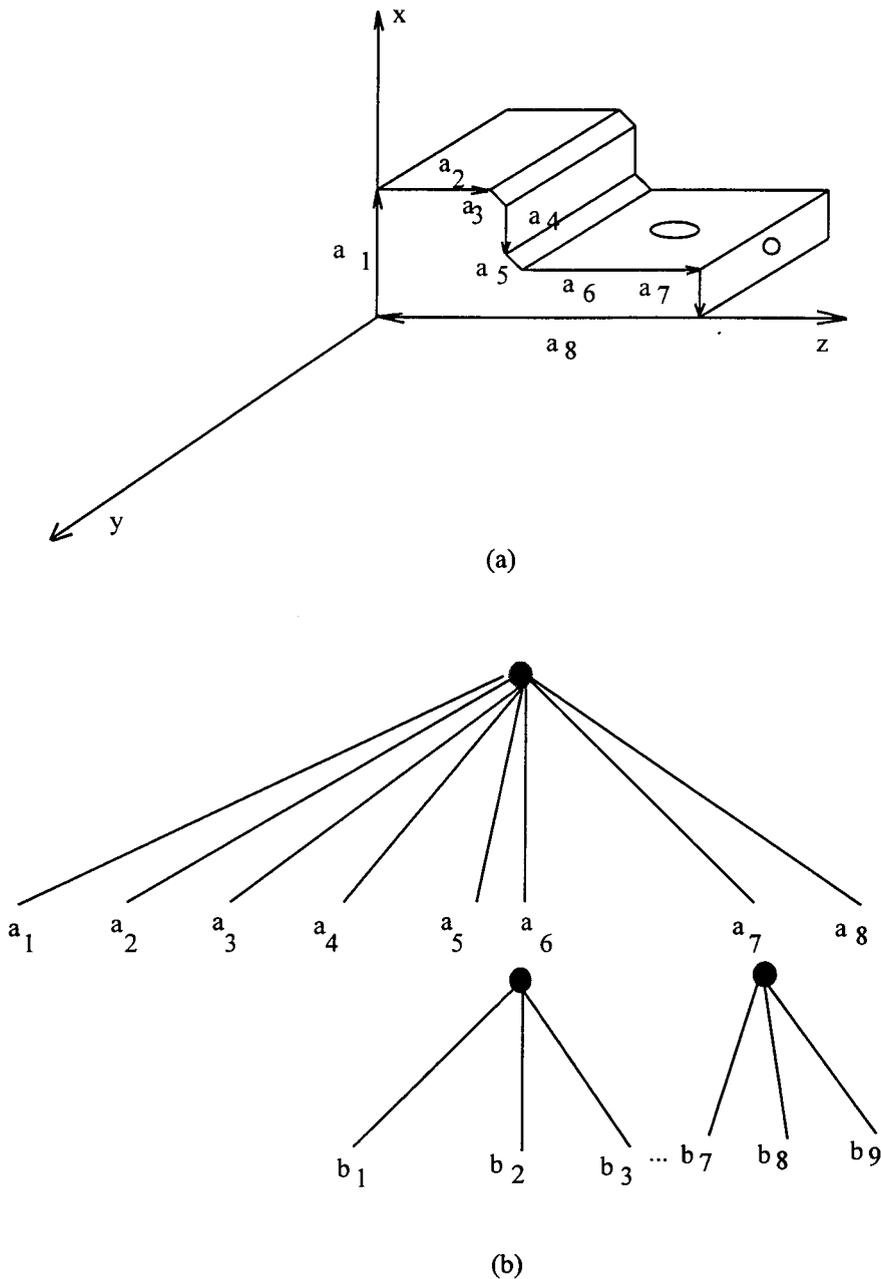


Figura 2.17: Representação de uma superfície (Srinivasan et al, 1985):

a) Sólido

b) Estrutura de Árvore

Wu et al (1986) aplicaram reconhecimento de padrão sintático para o projeto de células de manufatura. Os dados são relacionados às máquinas que produzem cada componente e identificados pelas letras "a" ou "b" ou "c", etc..., e cada peça é representada por uma string como "abc" ou "bcd", etc..., o qual indica a seqüência de processamento de diferentes componentes.

O autor utiliza o método da árvore de espalhamento mínimo para agrupar componentes com seqüência de processamento idênticos. O coeficiente de Levenshtein foi utilizado para determinar o menor número de transformações requeridas para derivar y de x, isto é, achar a distância entre o string "x" e "y". O MST é construído baseado nas distâncias entre todos os pares de strings.

Três tipos de transformações são utilizadas: deleção, inserção e substituição. Por exemplo, consideramos o string "abcd" e a sua distância aos strings "ab", "ade" e "ef" são:

abcd \rightarrow ab 2 eliminações

abcd \rightarrow ade 2 eliminações + 1 inserção

abcd \rightarrow ef 2 substituições + 2 eliminações

Dentro da abordagem de reconhecimento de padrões, ElMaraghy et al (1990) desenvolveram uma metodologia para alocar peças ao grupo de máquinas pela aplicação de alguns conhecimentos em sistemas especialistas, reconhecimento de padrões e conjunto difuso. Este método é baseado no conhecimento adquirido das características de projeto e processo de manufatura dos componentes. As sentenças simbólicas que contêm atributos de projeto e processamento dos componentes são analisadas pelo uso de regras de conhecimento (produção) e técnicas de reconhecimento de padrões de sintaxe, e, desta maneira, alocar as peças ao grupo de máquinas. O módulo esperto (analyser e o synthesizer) foi desenvolvido para alcançar o objetivo primordial de alocar peças ao grupo de máquinas. Com esta pesquisa eles tentam integrar CAD e CAM como também o problema de atribuir peças à célula de manufatura.

2.7 Sistemas Especialistas

Sistemas Especialistas são programas computacionais destinados a solucionar problemas em campos específicos do conhecimento. Sua principal

característica é a base de conhecimento que se refere ao domínio restrito em que o problema se insere.

Uma fase crucial do desenvolvimento de Sistemas Especialistas é a representação do conhecimento. Em geral, utiliza-se regras de produção, redes semânticas ou frames.

Sistemas Especialistas têm sido utilizados para resolver os mais variados problemas e aplicados às mais diferentes áreas, e é uma das maneiras de solução para o problema de TG. Um dos artigos apresentados nesta área foi proposto por Kusiak (1987, 1988) e resolve o problema da formação de células de manufatura pela utilização de um sistema especialista combinado com um algoritmo de agrupamento heurístico.

A metodologia procura agrupar máquinas e família de peças através de um algoritmo de agrupamento e quatro restrições relacionadas com :

- a capacidade de processamento de cada máquina;
- o tempo de processamento de cada máquina;
- requisitos de movimentação de material e máquinas para cada célula;
- restrições tecnológicas.

No modelo proposto, as restrições tecnológicas são apresentadas utilizando o sistema baseado em conhecimentos.

2.8 Análise de Agrupamento e Conjunto Difuso

Nos últimos anos vem desenvolvendo-se bastante pesquisa em lógica difusa, bem como aplicando seus conhecimentos no campo da pesquisa e desenvolvimento, em especial na área de administração da produção (Li et al, 1988; Xu et al, 1989).

A teoria dos conjuntos difusos apresenta como um dos seus conceitos fundamentais a noção de que os elementos podem pertencer ao conjunto com maior ou menor intensidade (Chu C. e Hayya J., 1991).

Contudo, no mundo real existem muitas situações onde não se vê com clareza a fronteira que separa uma coisa de outra; mas sim "regiões" de imprecisão. Conjuntos difusos são todos aqueles conjuntos no ambiente de informações incertas ou inexatas, isto é, todas aquelas coleções de objetos com fronteiras pouco nítidas para definir se pertencem ou não a uma dada classe (Chu C. e Hayya J. 1991). O conjunto difuso é caracterizado por um grupo de

elementos e uma função de pertinência definida por um número real no intervalo $[0,1]$. Por exemplo, um elemento x com grau de pertinência 0,8 significa que x é um membro de um dado conjunto com grau 0,8 numa escala onde "1" significa pertence completamente e "0" não pertence, concluindo que o elemento x mais pertence do que não pertence.

Um método baseado na análise de agrupamento difuso foi desenvolvido por Li Ding e Lei (1986), no qual ele usa as características de projeto na formação de família de peças. Ele utiliza uma função associada para descrever algumas características (comprimento da peça, por exemplo) difíceis de serem descritas num sistema binário. Ele adota um plano de classificação semelhante ao que usou Peklenik (Peklenik, J. e Grum, J. 1980). O sistema de classificação é feito em duas partes: primeiramente, é formada uma família de peças através da aplicação do algoritmo de agrupamento. Cada família é definida por uma peça referência ou um centróide cujas as coordenadas são os valores médios das coordenadas dos pontos pertencentes ao agrupamento. Num passo seguinte, um novo componente é alocado dependendo do grau de similaridade entre esta peça e cada peça referência. Este método, segundo os autores, é mais flexível do que o método apresentado por Peklenick (Peklenick e Grum 1980).

Outro trabalho foi desenvolvido por J.Y. Zhu e YU.Z. Zhang (1987), onde é utilizado a lógica difusa. Neste trabalho ele introduz os princípios básicos do método das relações de equivalência difusa (fuzzy equivalent relation method) para a formação de família de peças.

Ele constrói uma matriz cujos elementos são as características padrões de fabricação da peça (por exemplo, diâmetro interno, externo, rosca, etc...). Como cada uma destas características pode ter uma influência diferente na classificação foi associada a cada uma delas um peso. Este peso quantifica a influência daquela característica na classificação da peça. Após isto, ele usa um coeficiente de compatibilidade difusa, o qual indica a que família pertence cada peça.

2.9 Redes Neurais na Formação de Células de Manufatura

Redes neurais é uma teoria de simulação baseada no trabalho original de McCulloch e Pitts (1943), que estudou o funcionamento dos neurônios humanos. Nos anos 50 iniciaram-se os primeiros passos no desenvolvimento da Inteligência Artificial, e portanto estes conhecimentos seriam como fundamento. Redes neurais usam a idéia de neurônios e sinapses. Sabemos através da neurobiologia que os neurônios e sinapses são

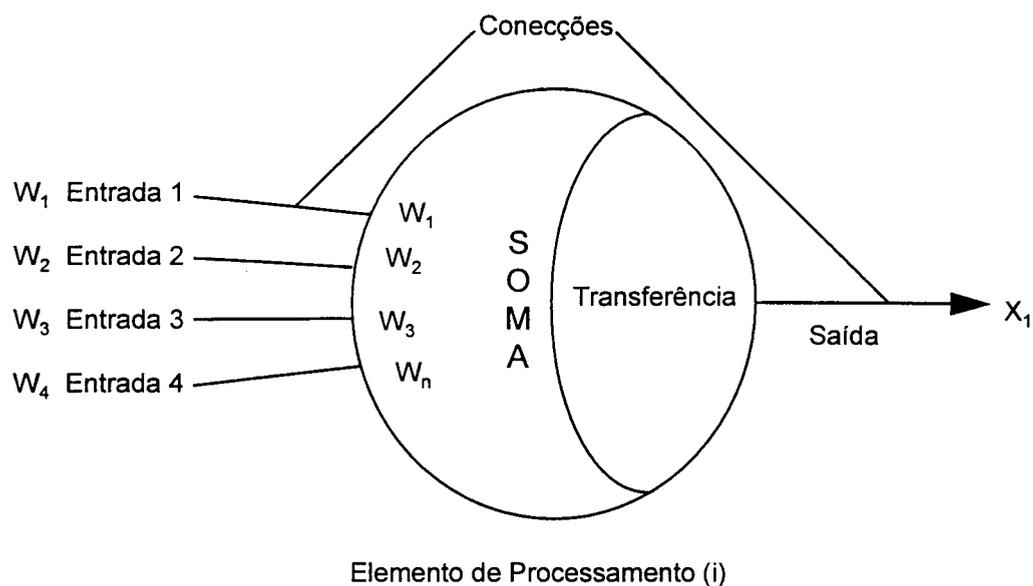
altamente interconectados, formando uma rede complexa de comunicação. Desta forma, redes neuronais são sistemas computacionais que envolvem um elevado número de elementos interconectados, que processam informações entre si. Os elementos básicos constituintes das redes neuronais são os elementos que processam a informação e as conexões entre eles (figura 2.18).

Nesta figura 2.18 (Klimasauskas, Casimir C., 1988), o círculo central representa um elemento de processamento da informação, que pode ser fornecido a ele por um número qualquer de conexões. Nesta mesma figura também está indicado como o neurônio processa esta informação. Isto é, cada informação recebe consigo um peso modulador, o que traduz ao neurônio a influência daquele sinal no conjunto de dados recebidos.

A rede assimila esse conjunto de informações e aprende como manejá-lo, criando com isto uma habilidade contínua de aprendizado e memória do sistema, podendo inclusive manipulá-lo e reorganizá-lo. Vemos que a Inteligência Artificial e redes neuronais são campos comuns de pesquisa. Modelos altamente sofisticados desta rede podem ser estudados nos trabalhos de Hopfield, Kohonen, Grossberg, etc...(Ali et al, 1990; Lippmann, 1987).

No tocante ao objetivo do presente estudo examinaremos o trabalho de Kusiak, A. e Chung, Y. (1991), que desenvolveram uma metodologia para a formação de células de manufatura aplicando Redes Neuronais associando a ela o conceito de ART - Adaptive Resonance Theory (Carpenter e Grossberg, 1987). Um algoritmo de aprendizado para o sistema peça-máquina foi desenvolvido, resultando em uma matriz contendo as peças gargalos, as quais podem ser removidas duplicando alguma máquina. Não há a intenção aqui de descrever em detalhes a forma pela qual o modelo GT/ART atinge seus objetivos, mas sim trazer para o campo de discussão mais um método disponível para a obtenção dos mesmos objetivos.

As regras de treinamento ou aprendizado estabelecem um conjunto de pesos iniciais, e indica como esses pesos deveriam ser adaptados durante o uso para melhorar o desempenho, como mostrado na figura 2.18. A habilidade de adaptação ou o contínuo aprendizado é o maior enfoque das pesquisas em redes neuronais



Neurodinâmica:

Função Soma:

$$I = W_0 + 1 W_1 X_1 + \dots + W_n X_n$$

Função Transferência

$$F(k) = (1 + e^{-k})^{-1}$$

Função Saída

$$X_1 = F(I)$$

Figura 2.18: Representação matemática de um elemento em processamento e suas conexões.

CAPÍTULO 3

3. - Algoritmo Genético

O Algoritmo Genético é um método de procura baseado na genética, isto é, num mecanismo de seleção natural. Novas gerações são criadas através de sobrevivência dos mais aptos e troca de informações aleatórias.

Sempre que procuramos um sistema com desempenho robusto, o algoritmo genético mostrou-se, através de muitas aplicações, principalmente nas áreas de engenharia, economia e administração ser um forte concorrente

O Algoritmo Genético foi proposto por Holland nos anos 70, e é um método que consiste em determinar uma solução ótima para problemas práticos envolvendo uma população de elementos. A idéia principal vem da maneira como a vida é reproduzida. Conforme sabemos, a manutenção da vida em nosso planeta baseia-se na forma como o RNA e o DNA trocam entre si seus códigos genéticos no processo de reprodução. Geração a geração, a vida vem sendo lentamente reproduzida, eventualmente por intermédio de mutações ou cruzamentos de seus gens dando origem a novos indivíduos, os quais devem ser testados por intermédio de uma função de ajuste (fitness). Caso eles sejam melhor adaptados eles sobrevivem e passam para a nova geração, senão são destruídos. Cria-se, assim, um indivíduo mais adaptado. Vemos, portanto, que é necessário a existência de um teste, o que é executado por uma função de ajuste (fitness). Esta é a idéia fundamental do Algoritmo Genético, ou:

- a) necessita de uma população inicial de indivíduos(cromossomos);
- b) necessita de uma maneira de reproduzir (aleatoriamente) os indivíduos da população original. Esta reprodução é tal que novos indivíduos são criados através do cruzamento e mutação;
- c) uma avaliação ou função de ajuste (fitness) para selecionar os melhores indivíduos.

3.1 Uma Comparação entre o Algoritmo Genético e os Métodos Tradicionais

De maneira a mostrar a soberania do Algoritmo Genético em relação aos outros métodos, algumas considerações deverão ser feitas em relação a essas diferenças.

Os aspectos a serem considerados são:

- I. conjunto de variáveis do problema e não com as variáveis físicas do mesmo;
- II. o Algoritmo Genético trabalha com uma população de variáveis codificadas do problema e não com um único valor da função (seja ela de que forma for);
- III. o Algoritmo Genético funciona baseado em melhor adaptabilidade (avaliação da função objetivo) não necessitando de derivadas ou de qualquer outro conhecimento funcional;
- IV. o Algoritmo Genético relaciona, cria, modifica variáveis de forma probabilística e não determinística.

Ao propormos um Algoritmo Genético, devemos, num primeiro estágio, criar um código. Este código é criado para representar os indivíduos da população afim de que ela possa ser tratada matematicamente.

A forma mais comum de representar os indivíduos tem sido o uso do código binário computacional, onde qualquer indivíduo de uma população pode ser representado por uma seqüência de zeros (0) e uns (1). Como ilustração mostraremos o processo proposto por Goldberg (1989), no qual ele usa o Algoritmo Genético. Vamos otimizar a função $f(x) = x^2$ no intervalo inteiro $[0,31]$. O máximo da função é conhecido, e é dado por $\max \{f(x)\} = f(31) = 961$, que é um dos extremos do intervalo de 0 até 31.

A figura 3.1 nos mostra que o máximo desta função encontra-se em $x=31$ da variável independente.

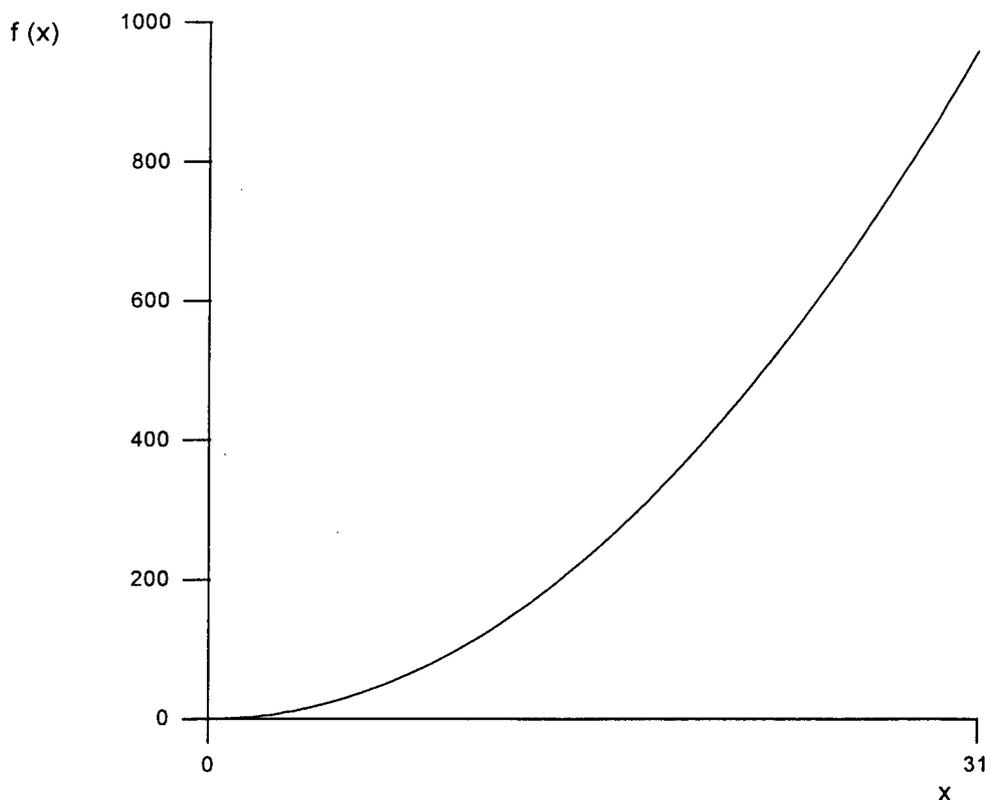


Figura 3.1: Gráfico da Função $f(x) = x^2$

Consideremos um sistema de chaves liga-desliga como mostrado na figura 3.2 onde o numeral "1" significa liga e o "0" desliga. A cada conjunto de posições de chaves recebemos um sinal de saída, $f(s)$.

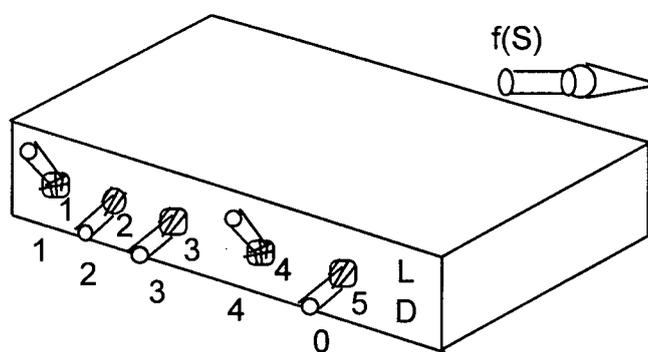


Figura 3.2: Sistema de Chaves para Obtenção do Sinal $f(s)$

A representação no sistema binário dos extremos dos intervalos $[0,31]$ será dada por $[00000; 11111]$ e a cadeia irá possuir cinco símbolos, os quais serão:

$$\{ \} 2^4 + \{ \} 2^3 + \{ \} 2^2 + \{ \} 2^1 + \{ \} 2^0 = x,$$

onde x é o valor da variável
no sistema decimal.

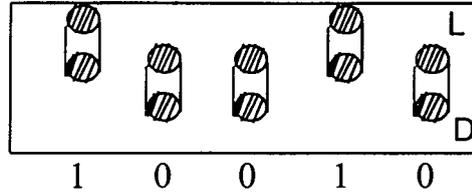


Figura 3.3: Posição das Chaves para a Cadeia 10010 = 18

Para melhor ilustrar, vejamos a figura 3.3 acima, onde a posição das chaves representa o elemento 10010, cujo valor decimal será computado como:

$$x^A \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 17^A$$

Para este valor decimal, o valor da função saída $f(s)$ é $f(17)=269$.

A grande diferença do Algoritmo Genético em relação aos métodos tradicionais é que ele não trabalha com a variável do problema, mas sim com uma representação codificada da mesma. É esta a chave do processo do Algoritmo Genético.

Como visto, nos processos otimizantes tradicionais procura-se o máximo ou o mínimo num determinado ponto e através de regras que nos fornecem a direção de crescimento da função calculamos novos pontos até atingirmos o ótimo da função.

Algumas funções são multimodais e às vezes os valores encontrados não correspondem ao ótimo esperado.

Ao contrário dos métodos tradicionais, o Algoritmo Genético trabalha com uma população de pontos e a procura do ponto ótimo usando uma população de valores da variável independente é mais eficiente.

Os métodos tradicionais utilizam critérios determinísticos, enquanto que o Algoritmo Genético utiliza regras totalmente probabilísticas nas suas alterações.

Podemos resumir as características do Algoritmo Genético da seguinte forma:

- 1 - usa um código;
- 2 - procura uma população;
- 3 - usa cegamente informações auxiliares;
- 4 - as operações são randômicas.

3.2 Aplicação Prática do Algoritmo Genético

Retornando ao problema anterior proposto por Goldberg (1989), vamos ver como um problema prático pode ser solucionado, ou melhor, simulado, usando Algoritmo Genético.

O problema apresentado é achar o máximo da função $f(x) = x^2$ no intervalo $[0; 31]$. O método é simples, sendo a idéia básica de se criar uma cadeia de caracteres formando a população, e depois aprimorar a população trocando alguns caracteres entre si.

Apresentaremos a seguir a seqüência de procedimentos:

- a) criação de uma população aleatória inicial usando uma moeda não viciada para criar cada indivíduo, contendo quatro deles, tais como:

01101, 11000, 01000, 10011;

- b) criação de novos indivíduos a partir da população inicial verificando se ela possui indivíduos melhores da que a anterior.

Na evolução das espécies, as regras fundamentais são:

1. reprodução de indivíduos;
2. cruzamento de parte dos cromossomos em torno de pontos específicos do DNA;
3. mutação, ou seja, numa posição aleatória alteração do tipo de um par de bases ao longo do DNA.

No caso de Algoritmo Genético, estas regras devem ser criadas para cada caso. Na reprodução, cada cadeia reproduz-se igual a si mesma, dentro de um certo critério, procurando otimizar a função objetivo de maneira a selecionar os indivíduos mais aptos.

Existem várias maneiras de implementar um sistema de reprodução, sendo que a mais simples é a criação de uma roleta, como mostrada na figura 3.4, onde cada fatia da roleta representa a maior adaptabilidade de um indivíduo, o qual depende proporcionalmente de sua aptidão de sobrevivência.

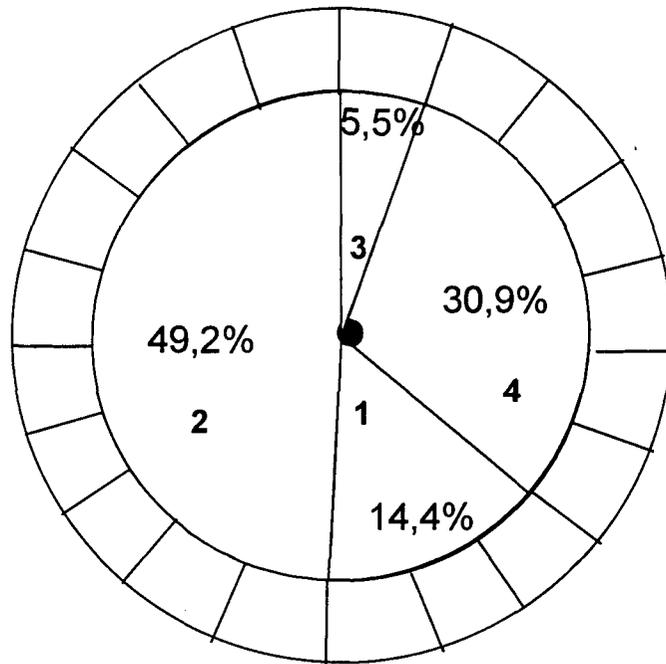


Figura 3.4 Roleta de Aptidões de Sobrevivência

A partir da primeira geração, colocando estes indivíduos numa tabela, como a tabela 3.1, calculamos o valor da função objetivo e o valor percentual de cada um destes valores.

Tabela 3.1: Valor da Função de Objetivo da Primeira Geração

Nº	Cadeia	X	Função Objetivo	% do Total
1	01101	13	169	14,4
2	11000	24	576	49,2
3	01000	8	64	5,5
4	10011	19	361	30,9
Total			1170	100,0

Examinando esta tabela podemos ver que os melhores valores são aqueles dados pelas cadeias 11000 e 10011 (cromossomo) onde as proporções no somatório de 49,2% e 30,9% corresponderão a fatias maiores da roleta, sendo que toda vez que girarmos a roleta haverá uma maior probabilidade dela parar nestes intervalos. Sempre que necessitamos selecionar elementos para a população giramos a roleta e obtemos um candidato, sendo que valores maiores

da função objetivo possuem, como visto, uma fatia maior na roleta, portanto uma maior chance de contribuírem com maior número de elementos.

Para se criar novos indivíduos, após o resultado da coleção de indivíduos mais aptos, devemos de forma probabilística selecionar os pares de cadeia que trocarão elementos entre si. O "crossover" ou cruzamento consiste, basicamente, na troca de partes completas das cadeias entre si, sendo necessário que os dois cromossomos possuam o mesmo número de elementos e que o cruzamento seja realizado no mesmo ponto em ambos. Como é escolhida a posição a partir da qual se efetuará a troca? O ponto é escolhido aleatoriamente entre o primeiro e o penúltimo elemento da cadeia, isto é, entre $[1, L-1]$, onde L é o número de elementos na cadeia. Por exemplo, se sortearmos o elemento K , então trocamos os elementos entre $K+1$ e L , ou seja, como a cadeia tem $L=5$, sortearmos um número entre 1 e 4.

Nos pares A_1 e A_2 e A_3 e A_4 é aplicado o crossover, isto é, a troca do último elemento entre as duas cadeias, originando o seguinte resultado:

$$A_1 = 0110||1 \Rightarrow 01100 = A'_1$$

$$A_2 = 1100||0 \Rightarrow 11001 = A'_2$$

Podemos dizer que o crossover desempenha um papel fundamental no processo de multiplicação, inovação e melhoria na evolução de uma população. No processo natural é através da mutação que é corrigida ou verificada a perda da potencialidade dos cruzamentos. Mas o que vem a ser a mutação no Algoritmo Genético? A mutação é a simples troca de um elemento de uma cadeia isolada, isto é, "0" por "1" ou vice-versa, e é rara e ocasional no mundo real e exerce um papel secundário no processo. Geralmente ela é destrutiva e somente ocorre com uma probabilidade baixíssima da ordem de 1 em 1000, quando então ela é evolutiva.

Podemos remontar a Tabela 3.1 do nosso exemplo originando a Tabela 3.2 de maneira a examinar algumas características importantes na formulação do Algoritmo Genético.

Tabela 3.2: Exemplo de Aplicação do Algoritmo Genético

População Inicial						
Cadeia No	População Inicial	x Decima l	$f(x) = x^2$	$\frac{f(x)}{\sum f(x)}$	$\frac{f(x)}{\overline{f(x)}}$	Sorteio da Roleta
1	01101	13	169	0,14	0,58	1
2	11000	24	576	0,49	1,97	2
3	01000	8	64	0,06	0,22	0
4	10011	19	361	0,31	1,23	1
Soma $\sum f(x)$			1170	1,000	4,00	4,00
Média $\overline{f(x)}$			293	0,250	1,00	1,00
Máximo			576	0,490	1,97	2,00

Verificamos nesta tabela que através da função objetivo $f(x) = x^2$ temos os indivíduos mais aptos, os quais são precisamente $x = 24$ e $x = 19$ com suas respectivas funções $f(x) = 576$ e $f(x) = 361$. No sorteio da roleta temos uma expectativa maior de obtermos o indivíduo '2', depois "4" e assim por diante "1" e "3".

Observando a tabela podemos verificar que quase todas as cadeias têm os elementos melhorados. Para continuar, nós eliminamos aqueles que não são aptos e prosseguimos até encontrarmos o máximo e mais nenhum melhoramento é obtido.

No nosso exemplo é criada a Tabela 3.3, que é a nova geração, obtida pelo acionamento da roleta, formada pelos elementos 01101 (uma vez); 11000 (duas vezes); 10011 (uma vez) seguida do crossover ou cruzamento, gerando assim uma nova população. Após o crossover ocorrer, pois a possibilidade de ocorrer em 20 lugares distintos é de $20 \times 0,001 = 0,02$.

Tabela 3.3 : Criação da Nova População

Criação da Nova População			
Conjunto para o "Cross-Over"	Par	Local do Cruzamento " "	Nova População
0110 1	2	4º (elemento)	01100
1100 0	1	4º (elemento)	11001
11 000	4	2º (elemento)	11011
10 011	3	2º (elemento)	10000

Outra vez podemos remontar a tabela 3.3 originando a tabela 3.4, a qual faz uma avaliação em termos de média do valor da função objetivo e do valor máximo da função objetivo da nova população criada.

Tabela 3.4 : Desempenho da Segunda Geração de Cadeias

Desempenho da Segunda Geração de Cadeias						
Cadeia No	Nova População	x Decima 1	$f(x)=x^2$	$\frac{f(x)}{\sum f(x)}$	$\frac{f(x)}{\overline{f(x)}}$	Sorteio da Roleta
1'	01100	12	144	0,082	0,328	0
2'	11001	25	625	0,356	1,425	1
3'	11011	27	729	0,416	1,662	2
4'	10000	16	256	0,146	0,583	1
Soma $\sum f(x)$			1754,0	1,000	4,000	4,00
Média $\overline{f(x)}$			438,5	0,250	1,000	1,00
Máximo			729,0	0,416	1,662	2,00

Verificando a tabela 3.4, notamos que as cadeias que começam com o par 11... têm maior número de indivíduos na população, o qual é uma qualidade dos indivíduos mais aptos, fornecendo com isto a definição de gabarito (schema) ou gabaritos (schemata), que é assunto do próximo item.

3.3 Gabarito e sua Importância na Criação do Código

Ao examinarmos as cadeias devemos olhá-las como estruturas organizadas e desta feita classificá-las como um arranjo de seus elementos. Para melhor elucidar, dizemos que qualquer cadeia é uma combinação de alfabeto {0; 1; *}, onde "*" asterisco significa posições na cadeia que não nos interessa. Um cromossomo que classifica o animal em seu filo no sistema biológico, o elemento "*" ou "tanto faz qualquer elemento", representa cor dos olhos, altura, forma das orelhas, cor da pele, etc.... Desta maneira, podemos representar as cadeias por {0; 1; *} de forma a compará-las à procura de semelhança. Por exemplo, se possuímos uma régua com determinados orifícios, aos quais são associado os numerais {0; 1; *} onde "*" pode assumir os valores 0(zero) ou 1(um).

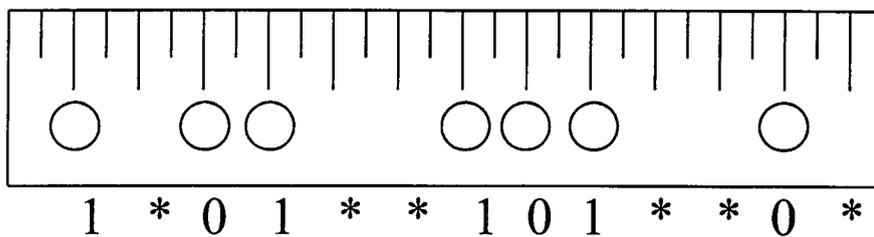


Figura 3.5: Régua Gabarito

A esta régua é que chamamos de gabarito (schema, schemata).

Para melhor esclarecimento de gabarito, suponhamos o conjunto de cinco cadeias com dez elementos cada um mostrado abaixo:

```

0 1 0 1 1 1 0 1 0 0
0 1 1 1 1 0 1 1 0 0
0 1 0 1 1 0 0 0 0 0
0 1 1 1 1 1 0 0 0 0
0 1 0 1 1 0 1 0 0 0

```

O gabarito então é da forma

```
[0 1 * 1 1 * * * 0 0]
```

Podemos, com isto, determinar o número de gabaritos criados para uma cadeia de "L" elementos dado um alfabeto de "K" elementos. No caso de $L = 5$ e $K = 2$, teremos $2^5 = 32$ gabaritos. No entanto se introduzirmos o metasímbolo "*" esta possibilidade passa a ser $(2+1)^5 = 243$ gabaritos, com um aumento substancial de indivíduos.

Os gabaritos ao participarem dos cruzamentos podem ser destruídos como nos exemplos $[0 * * * 1]$ e $[* * 0 1 *]$, onde o segundo terá mais chance de se multiplicar do que o primeiro.

3.4 A Sobrevivência dos Gabaritos Mais Aptos

As operações com o Algoritmo Genético, como já foi visto, são simples, onde com uma população de "n" indivíduos copiamos estes entre si, cruzarmo-los, mutamo-los de maneira a selecionar os mais aptos.

Devemos definir algumas notações que deverão ser utilizadas ao longo do capítulo. É usada a letra maiúscula para representar a cadeia (cromossomo) e os seus elementos (genes) aparecerão como letras minúsculas seguida, de um índice que indicará a sua posição na cadeia. Assim sendo,

$$A_1 = a_1 a_2 a_3 a_4 a_5 a_6$$

representará uma cadeia do tipo:

$$A = 0 1 0 1 1 0.$$

A cadeia deverá ser representada em função do tempo, ou seja, $A_1 = A_1(t)$, pois o processo de reprodução se desenvolve ao longo do tempo.

Sendo assim, a população total deverá ter a seguinte notação:

$$A(t) \Rightarrow \{A_1(t), A_2(t), \dots, A_n(t)\},$$

onde o índice "n" representa o número de indivíduos da população.

Para melhor compreendermos o comportamento dos gabaritos, o qual denotamos por H, é imprescindível definirmos a sua ordem e o seu comprimento de maneira a classificá-los e verificar a função dos mesmos nos processos de reprodução, cruzamento e mutação.

A ordem de um gabarito H é representada por $O(H)$ e consiste no número de algarismos constante na cadeia, como por exemplo no gabarito $H = **011**0$, a ordem é $O(H) = 4$.

O comprimento de um gabarito $\delta(H)$ é definido como a diferença entre a posição do último e o primeiro elemento do alfabeto na cadeia, como por exemplo, no gabarito $H = **011**0$, o comprimento é definido como $\delta(H) = 8 - 3 = 5$.

As definições apresentadas servirão de base para mostrar a função dos gabaritos nos processos de reprodução, cruzamento e mutação.

Verifiquemos primeiramente o que ocorre no processo da reprodução de gabaritos. Seja uma população $[A(t)]$ de "m" gabaritos do tipo H num instante "t" representada por $m = m(H,t)$. Sabemos que um gabarito é copiado, durante o processo de reprodução, enviando uma cadeia " A_i " com probabilidade $P_i = f_i / \Sigma f_i$, logicamente obedecendo a função objetivo. A nova população de "m" indivíduos em substituição à população $A(t)$, é dada por:

$$m(H, t + 1) = \frac{m(H, t) \cdot n \cdot f(H)}{\sum f_i} \quad (3.1)$$

onde $f(H)$ é a média de adaptabilidade do gabarito H , e $\sum f_i / n$ é a média de adaptabilidade de toda população. A equação (3.1) poderá ser reescrita como:

$$m(H, t+1) = m(H, t) f(H) / \bar{f} \quad (3.2)$$

Ao interpretarmos a equação (3.2), verificamos que o gabarito cresce proporcionalmente à razão entre a média de adaptabilidade dos indivíduos e da população, isto é, se a média de adaptabilidade dos indivíduos for maior do que a média de população ele terá probabilidade com sucesso de sobreviver.

Esta propriedade nos mostra que não precisamos trabalhar com uma cadeia inteira, mas sim com um gabarito com média de adaptabilidade mais elevada, pois estas informações são transmitidas de geração a geração. Suponhamos que um determinado algoritmo sobreviva acima da média num processo de reprodução, ou seja, $c \times \bar{f}$ e a partir da equação (3.2) teremos que:

onde:

$$f(H) = \bar{f} + c\bar{f}$$

$$m(H, t + 1) = m(H, t) \left[\frac{\bar{f} + c\bar{f}}{\bar{f}} \right] \quad (3.3)$$

simplificando, teremos:

$$m(H, t + 1) = (1 + c)m(H, t) \quad (3.4)$$

Analisando a equação (3.4), verificamos que na próxima geração o número de cadeias contidas no gabarito H é "c" vezes maior do que a anterior, e para uma seqüência de eventos num intervalo [0,t] podemos reescrever a equação (3.4) como:

$$m(H, t) = m(H, 0)(1 + c)^t \quad (3.5)$$

A equação (3.5) nada mais é do que a equação de juros compostos e nos indica o crescimento exponencial do número de gabaritos com média de adaptabilidade acima da análise da população, provocando, assim, a tendência de que os menos aptos serão eliminados do processo.

A nossa meta, logicamente, é encontrar a melhor solução para a função objetivo e, como vimos, a reprodução tem por si só a escolha dos mais aptos. Assim surge a seguinte pergunta:

Como alcançar a melhor solução para a função objetivo?

Esta pergunta é respondida pelo processo de cruzamento e mutação, o qual afeta a taxa de crescimento e decrescimento exponencial dos gabaritos.

Passaremos agora a analisar como a função de cruzamento pode afetar um gabarito. Consideremos a seguinte cadeia de elementos:

$$A = 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1$$

Nesta cadeia de 7 (sete) elementos, podemos obter dois gabaritos, $H_1 = * * 0 * * * 1$ e $H_2 = * * * 0 1 * *$, os quais são cruzados numa posição 3(três) escolhida aleatoriamente. A representação do corte e o comprimento dos gabaritos serão:

$$\begin{aligned} A &= 0 \quad 1 \quad 0 \quad || \quad 0 \quad 1 \quad 0 \quad 1 \\ H_1 &= * \quad * \quad 0 \quad || \quad * \quad * \quad * \quad 1 \quad \therefore \quad \delta(H_1) = 7 - 3 = 4 \\ H_2 &= * \quad * \quad * \quad || \quad 0 \quad 1 \quad * \quad * \quad \therefore \quad \delta(H_2) = 5 - 4 = 1 \end{aligned}$$

Ao verificarmos os gabaritos H_1 e H_2 , vemos que o gabarito H_1 , dificilmente se manterá sendo provavelmente destruído após o corte, o mesmo não ocorrendo com H_2 . Sendo assim, o gabarito de menor comprimento terá menor probabilidade de o corte cair entre os seus elementos.

Portanto, a probabilidade de destruição dos gabaritos é dada por:

$$p_d(H) = \frac{\delta(H)}{(L-1)} \quad (3.6)$$

onde:

$p_d(H)$ = probabilidade de destruição;
 $\delta(H)$ = comprimento do gabarito;
 L = número de elementos na cadeia.

Aplicando a fórmula ao nosso exemplo, teremos que $p_d(H_1) = 4/6$ e $p_d(H_2) = 1/6$, mostrando matematicamente a dificuldade que o gabarito H_1 terá de sobreviver a um cruzamento.

Sendo p_c a probabilidade de ocorrer cruzamento com determinado indivíduo (gabarito), podemos definir a probabilidade de sobrevivência deste gabarito como:

$$p_s(H) \geq 1 - p_c \left[\frac{\delta(H)}{(L-1)} \right] \quad (3.7)$$

onde:

$p_s(H)$ = probabilidade de sobrevivência;
 $p_c(H)$ = probabilidade de cruzamento;
 $\delta(H)$ = comprimento do gabarito;
 L = número de elementos na cadeia.

Se supormos que temos uma população de m ($H < t$) gabaritos num determinado instante t , a equação (3.7) será representada após a reprodução com cruzamento num tempo $t + 1$, por:

$$m(H, t + 1) \geq m(H, t) \left[\frac{f(H)}{f} \right] \left\{ 1 - p_c \left[\frac{\delta(H)}{L-1} \right] \right\} \quad (3.8)$$

Ao examinarmos a equação 3.8, dois pontos devem ser destacados, aos quais influem no crescimento do número de gabaritos nas novas gerações:

1. se o gabarito tem média de adaptabilidade maior ou menor do que a média da população;
2. se ele possui um comprimento maior ou menor.

Como conclusão, podemos dizer que os gabaritos com adaptabilidade acima da média da população e com comprimento menor terão maiores possibilidades de sobreviver e com isto aprimorar o Algoritmo Genético.

Por fim, verificamos a influência da mutação na taxa de crescimento dos gabaritos. Sendo as mutações individuais independentes entre si e seja P_m a probabilidade de mutação, definimos a probabilidade de sobrevivência de um gabarito H de ordem $O(H)$ como:

$$(1 - P_m)^{O(H)} \quad (3.9)$$

onde:

P_m = probabilidade de mutação;
 $O(H)$ = ordem do gabarito;

sendo que P_m é um valor muito pequeno. Assim sendo, a equação (3.9) poderá ser escrita como

$$(1 - P_m)^{O(H)} \cong 1 - P_m \times O(H) \quad (3.10)$$

Reescrevendo a equação (3.8), levando agora em consideração o processo de mutação no processo de reprodução de gabaritos num tempo $t+1$, teremos:

$$m, (H, t + 1) \geq m(H, t) \left[\frac{f(H)}{f} \right] \left\{ 1 - P_c \left[\frac{\overset{\delta}{d}(H)}{(L-1)} \right] - O(H)P_m \right\} \quad (3.11)$$

A parcela $O(H)P_m$ pouco contribui no resultado, mas é utilizada na medida em que pretendemos renovar o processo de criação.

Como vimos, a potencialidade do Algoritmo Genético em comparação com os métodos tradicionais, assim como sua simplicidade matemática, nos faz repensar quão prodigiosa é a teoria da evolução das espécies e como ela pode ser aplicada nos processos de otimização.

3.5 Aplicação Prática dos Teoremas

Retornando ao item 3.2 é apresentado o problema de maximização da função $f(x) = x^2$ no intervalo $x \in [0,31]$. Verificando a Tabela 3.2 e 3.3 e selecionando 3 (três) gabaritos da cadeia com valores melhores da função objetivo, teremos

$$A_2 = 11000 \Rightarrow \begin{cases} H_1 = 1 * * * * \\ H_2 = * 1 0 * * \\ H_3 = 1 * * * 0 \end{cases} \quad (3.12)$$

Dada a cadeia e a população inicial de gabaritos, observemos a seguir o efeito da reprodução, cruzamento e mutação sobre essa população.

Começaremos a nossa análise pelo gabarito H_1 em relação à ocorrência de reprodução com cruzamento e verificamos que esse gabarito está presente nas cadeias A_2 e A_4 da Tabela 3.2. O mesmo deverá ser feito para os gabaritos H_2 e H_3 e assim sendo, podemos construir a Tabela 3.5 onde deverá constar também a média do gabarito antes da reprodução.

Tabela 3.5: Formação dos Gabaritos

Gabaritos para o Problema $f(x) = x^2$ Antes da Reprodução		
Gabarito	Cadeias que contêm os Gabaritos	Média do Gabarito
$H_1 = 1 * * * *$	2 e 4	$\left(\frac{576 + 361}{2}\right) = 469$
$H_2 = * 1 0 * *$	2 e 3	$\left(\frac{576 + 64}{2}\right) = 320$
$H_3 = 1 * * * 0$	2	576

A tabela 3.6 é construída levando em conta o desempenho do gabarito, primeiramente após a reprodução e numa segunda etapa após as operações realizadas, tais como reprodução, cruzamento e mutação.

Tabela 3.6 :Gabaritos da Nova População

Gabaritos para o Problema $f(x) = x^2$					
Depois da Reprodução			Após todas as operações		
Nº de Gabaritos após Contagem	Valor Real	Cadeias que contêm os Gabaritos	Nº da Contagem Esperada	Valor Real	Cadeias nas quais os Gabaritos estão
3,20	3	2, 3, 4	3,20	3	2, 3, 4
2,18	2	2, 3	1,64	2	2, 3
1,97	2	2	0,0	1	4

Aplicando o teorema do crescimento do número de gabarito na nova geração, o valor esperado para os gabaritos H_1 , H_2 e H_3 será dado por:

$$m(H_1, t+1) = 2 \cdot \frac{(576 + 361) / 2}{293} = 3.197 \approx 3.20$$

$$m(H_2, t+1) = 2 \cdot \frac{(576 + 64) / 2}{293} = 2.184 \approx 2.18$$

$$m(H_3, t+1) = 1 \cdot \frac{576}{293} = 1.9658 \approx 1.97$$

Examinando a tabela 3.5 e 3.6, verificamos que $m(H_1, t) = 2$ na população inicial e que após a reprodução ele permanece em três cadeias na nova população, o que concorda plenamente com o valor esperado, $m(H_1, t+1) = 3,2 \approx 3.0$.

Aplicando os teoremas (equação 3.9), isto é, levando em consideração somente a reprodução e crossover, pois a probabilidade de mutação é muito pequena ($P_m = 0,001$), e sendo:

$$\begin{array}{lll} m(H_1, t) = 2 & \overset{\delta}{\partial}(H_1) = 0 & f(H_1) = 469 \\ m(H_2, t) = 2 & \partial(H_2) = 3 - 2 = 1 & f(H_2) = 320 \\ m(H_3, t) = 1 & \partial(H_3) = 5 - 1 = 4 & f(H_3) = 576 \\ \bar{f} = 293 & P_c = 1 & L - 1 = 5 - 1 = 4 \end{array}$$

teremos que:

$$m(H_1, t+1) = 2.0 \times \frac{469}{293} \left(1 - \frac{0}{4}\right) = 3.20$$

$$m(H_2, t+1) = 2.0 \times \frac{320}{293} \left(1 - \frac{1}{4}\right) = 1,64$$

$$m(H_3, t+1) = 1.0 \times \frac{576}{293} \left(1 - \frac{4}{4}\right) = 0$$

Assim podemos concluir que:

- a) não é possível aumentar mais o número de cadeias contendo o gabarito H_1 ;
- b) o gabarito H_2 permanece em duas cadeias;
- c) o gabarito H_3 permanece em uma cadeia.

Com este simples exemplo podemos exemplificar o teorema onde mostra o crescimento de forma exponencial dos gabaritos mais aptos e o decréscimo dos menos aptos.

3.6 Uma retrospectiva sobre Algoritmo Genético

Como já foi visto, o Algoritmo Genético é uma técnica de otimização que procura imitar a seleção natural e o processo de evolução biológico dos seres vivos.

O Algoritmo Genético tem sido aplicado com bastante sucesso nos mais diferentes problemas de otimização, tais como o problema do caixeiro viajante (OLIVER, I. M. et al, 1987) e o sequenciamento no job shop (BIEGEL, J. e DAVERN, J., 1990). Os pesquisadores Gynar E. Liepins e W. D. Polter refletem nos seus trabalhos através do seu estudo sobre Diagnóstico de Falta Múltipla a interface entre a Pesquisa Operacional e o Algoritmo Genético (DAVIS, L., 1991). Yuval Davidor mostrou como o Algoritmo Genético é aplicado na geração da trajetória do robô.

Uma série de pesquisas tem sido realizadas no campo de Redes Neurais e a sua interação com o Algoritmo Genético. Dentro destes artigos temos o apresentado por Steven A. Harp e Tariq Samad, que descreve técnicas de como codificar a arquitetura de redes neuronais em cromossomos binários (DAVIS, L., 1991).

Os autores David J. Powell, Michael M. Skolnick e Siu Shing comparam um número de técnicas de otimização em problemas de projeto. É utilizada uma técnica chamada de interdigitação, a qual intercala a otimização baseada em regras com o Algoritmo Genético (DAVIS, L., 1991).

Dentro da linha de sequenciamento temos o trabalho desenvolvido por Gilbert Syswerda (DAVIS, L., 1991). Ele procura descrever a aplicação do Algoritmo Genético para o sequenciamento de atividades num laboratório, discutindo também os fatores a serem considerados no domínio do sequenciamento. Daniel Whitley, Timothy Starkweather e Daniel Shaner

descreveram uma técnica para a solução do problema do caixeiro viajante, usando um problema de otimização combinatória com Algoritmo Genético .

Como vimos, uma variedade de aplicações de Algoritmo Genético foram realizados na área da Engenharia de Produção, tentando solucionar problemas do mundo real.

3.7 Conclusão

Todas estas idéias geradas no processo de desenvolvimento do Algoritmo Genético podem ser aplicadas aos mais diversos ramos do conhecimento humano. Neste trabalho criou-se um código de representação do cromossomo para o problema de formação de células de manufatura. As operações lógicas e instruções constituem-se nos elementos que formam o código e serão objeto de estudo nos capítulos subseqüentes.

CAPÍTULO 4

4. - O Planejamento do Processo

O planejamento do processo é uma função da Engenharia de Manufatura que produz planos de processo com base nas especificações do projeto de um certo produto. Consta do plano de processo a seleção das estações de trabalho, rota e o detalhamento das operações(velocidade, avanço, etc..). O projeto especifica os atributos físicos do produto, tais como dimensões, tolerâncias, material, acabamento, etc..., e também os atributos funcionais, isto é, o desempenho esperado.

É visto que, enquanto o projeto serve como entrada na função de planejamento de processo, o planejamento de processo restringe a ação do sistema de planejamento e o controle do chão de fábrica. Lógico que haverá um certo retorno do planejamento do processo para o projeto, tais como o processo desejável, quais as características de projeto que podem ser melhores exploradas e a manufaturabilidade do projeto do produto. Igualmente, há uma interação entre a função do planejamento do processo e as funções de planejamento e controle do chão de fábrica, de maneira que as máquinas gargalos sejam duplicadas.

O planejamento do processo é muito importante, pois a habilidade da resposta da manufatura, mudanças no projeto, introdução de novos produtos e mudanças na capacidade dos recursos dependem basicamente dele.

O planejamento de processo é um dos componentes básicos de todo o sistema de manufatura. A figura 4.1 mostra o fluxo de informações necessárias ao desenvolvimento de um planejamento do processo dentro de uma companhia.

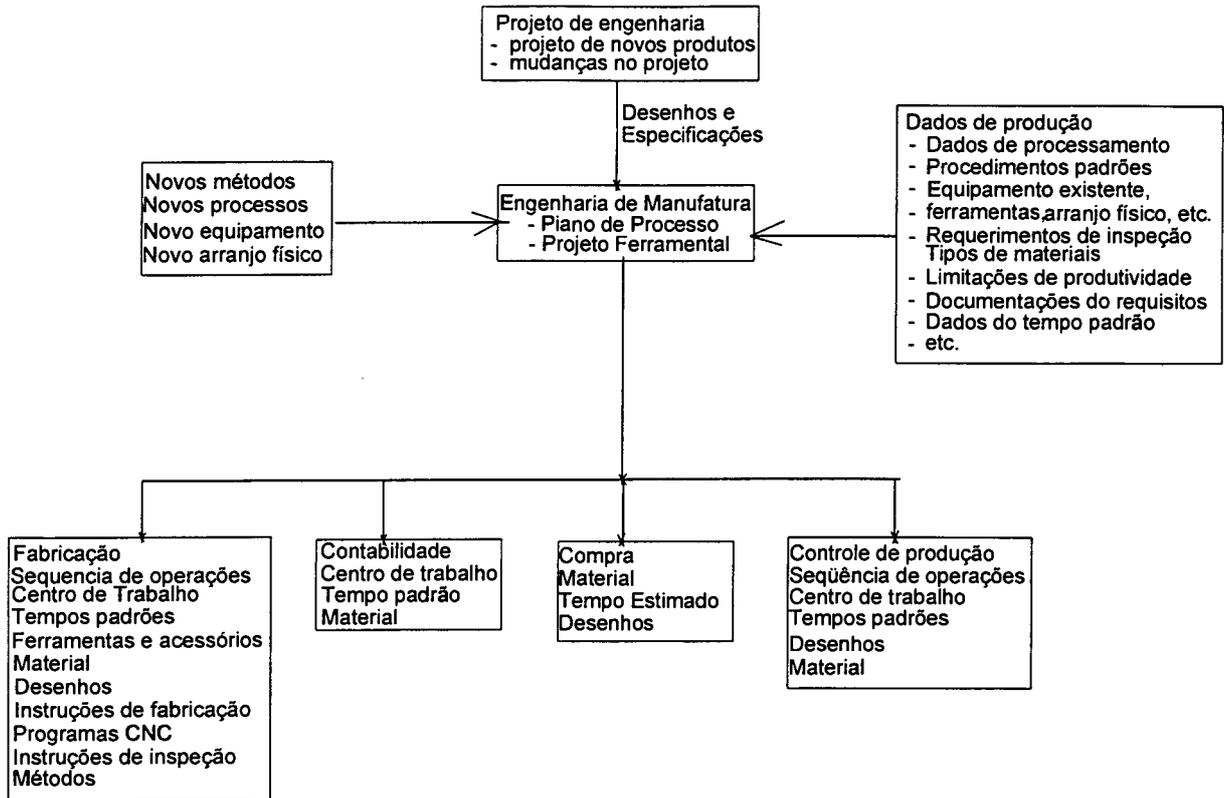


Figura 4.1: Fluxo de Informação do Planejamento do Processo (Bedworth et al, 1991).

Normalmente o planejamento de processo é feito manualmente pela maioria das companhias sem um procedimento formal. Cada vez que uma nova peça é projetada, um planejador olha o desenho do componente e decide qual máquina será utilizada, quais as operações serão realizadas e em qual seqüência.

Existem duas razões pela qual as companhias geralmente geram planos de processo em excesso. Primeiramente, dentro de uma companhia existem vários planejadores, provavelmente cada um apresentando um planejamento de processo diferente para a mesma peça. E por último, o planejamento do processo é descrito com relação à configuração das máquinas ferramentas em mente. Com a ajuda do Planejamento do Processo Auxiliado por Computador (CAPP) e Tecnologia de Grupo (TG) é possível padronizar tais planos, reduzir o número de novos planos de processo, assim como guardar, recuperar e editar planos com bastante eficiência (Bedworth et al, 1991).

O planejamento do Processo com CAPP é abordado de três diferentes maneiras: planejamento variante, semi-generativo e generativo.

O processo variante se baseia numa estrutura que faz o agrupamento de peças geometricamente semelhantes ou com processo de fabricação semelhantes. A estes grupos de peças chamados famílias será elaborada uma folha de processo, isto é, um plano padrão que deverá conter no

mínimo a seqüência das operações. Um plano padrão é desenvolvido para cada família de peças. Ele é guardado numa matriz de família de peças. Cada plano padrão poderá ser modificado de maneira a torná-lo exequível para uma específica peça, gerando dessa forma uma nova folha. Esta técnica de planejamento de processo utiliza a classificação e codificação das peças juntamente com o conceito de Tecnologia de Grupo.

O planejamento de processo variante apresenta-se em dois estágios: um preparatório e outro de produção. No estágio preparatório as peças são classificadas, codificadas e posteriormente agrupadas em famílias. Existe uma série de métodos de classificação e codificação vistos anteriormente como o OPTIZ (Alemanha); KK-1, KK-3, KC-1 (Japão); MICLASS (Holanda/EUA); DCLASS, CODE, COFORM (EUA).

O segundo estágio trata da execução de um novo projeto ou de uma nova peça. Esta peça deverá ser codificada e haverá uma busca no arquivo para tentar enquadrá-la em uma das famílias existentes. O plano padrão será resgatado e o operador deverá efetuar as adaptações do plano à nova peça.

Existem sistemas CAPP comerciais que utilizam o método variante, o qual exige que as peças estejam divididas em famílias de peças, as quais são agrupadas através da aplicação de teoria de Tecnologia de Grupo.

O problema crucial no sistema variante é que ele não gera novos planos de processos. Entretanto, o mesmo ainda domina nas indústrias, porque é fácil de ser implementado e pode abranger uma grande variedade de peças.

Devido à dificuldade de gerar novos planos de processo, um novo sistema foi desenvolvido chamado de planejamento de processo generativo.

No planejamento de processo generativo um plano individual é criado para cada peça. Baseado numa análise da geometria da peça, material e outros fatores, os quais influenciam as decisões de manufatura, o sistema gera um novo plano de processo para cada peça. Logo, o sistema generativo procura o desenvolvimento de um plano de trabalho (folha de processo) a partir de informações de projeto contidas numa base de dados, através de inferências automáticas ou iterativas. Ele deverá captar o conhecimento de manufatura e codificá-lo em em uma base de dados. No processo generativo, as funções como seleção de máquinas-ferramentas, ferramentas, etc..., devem ser feitas automaticamente.

As vantagens de processo generativo são (Bedworth et al, 1991):

- pode gerar planos consistentes rapidamente;
- aumenta a facilidade para produzir planos para novos componentes;
- pode receber e transmitir informações atualizadas.

Os sistemas atuais existentes não são totalmente generativos, pois seria necessário que todas as decisões como a capacidade dos processos e custos fossem administradas pelo software. Os sistemas em oferta que trabalham com método generativo foram desenvolvidos para resolverem problemas específicos de planejamento.

Alguns exemplos de sistemas generativos são AUTAP, ICAPP, TIPPS (ALTING LEO et al, 1989).

O sistema híbrido apresenta características do sistema variante e generativo. O sistema generativo não é totalmente flexível e o variante tem problemas de codificação, implantação e espaço de memória.

Elucidamos algumas vantagens do sistema híbrido (Bedworth et al, 1991):

- flexibilidade orientada conforme o tipo de produção do usuário;
- uso das potencialidades do sistema maximizado, uma vez que o sistema é desenvolvido especialmente para o usuário, levando em consideração as características de sua empresa;
- custo de implantação e operação reduzidos;
- sistema de fácil implementação visando crescer junto com a empresa;
- possibilidade de armazenamento de planos que forem de interesse da empresa;
- facilidade de recuperação de planos armazenados com possibilidade de alteração destes planos.

4.1 Planos de Processos Alternativos

A idéia de que uma peça pertencente a uma família permaneça somente num grupo de máquinas ou que uma máquina pertencente a um grupo que manufacture somente peças da mesma classe é uma ilusão. No meio industrial, a realidade mostra que a presença de planos de processos alternativos e duplicação de máquinas deverão ser levados em consideração na formulação do modelo.

Na formação da matriz e formulação do agrupamento, uma seqüência de operações num fluxo de produção de uma peça, não é levada em consideração, provocando um efeito adverso em termos de tempo e custos de movimentação de materiais.

Kusiak e Fink (1988) desenvolveram um modelo para a seleção de um conjunto de planos de processos na área de usinagem com o mínimo de máquinas. Esta metodologia aborda certas considerações fundamentais como a taxa de fluxo da peça, tempo e etapas do processamento, os quais são influenciados pela seleção do plano de processo.

4.2 Agrupamento Máquina-Peça em Presença de Planos de Processo Alternativos

A maioria dos problemas de formação de células de manufatura encontrado na literatura assume que cada operação de uma peça é restrita a uma única máquina. Esta informação como já visto é expressa na forma de seqüência de operações de cada peça ou na forma de matriz peça-máquina. Entretanto, num sistema de manufatura celular, uma operação pode ser realizada por máquinas alternativas, as quais têm a capacidade requerida para executar esta operação.

Kusiak (1987) mostrou que ao se considerar planos de processos alternativos resulta numa melhoria qualitativa na formação de família de peças e grupo de máquinas. Neste modelo não foi considerado o orçamento e a capacidade das máquinas. Choobinch (1988) desenvolveu um modelo aplicando programação inteira, no qual ele leva em consideração o orçamento, a capacidade das máquinas e o fato de que algumas operações podem ser realizadas em mais de uma máquina. Ele adotou para a formação da célula um método seqüencial e não considera explicitamente o plano de processo alternativo. É apresentado um método para resolver o problema de agrupar peças-máquinas em células, baseado na capacidade das máquinas para processar as peças em consideração. Os requerimentos para o processamento das peças são descritos em relação às necessidades ferramentais; e a capacidade da máquina é expressa em termos de ferramentas disponíveis em cada máquina e o tempo de processamento. Uma peça poderá ser fabricada em qualquer máquina que tenha as ferramentas necessárias para realizar esta operação.

A metodologia usada no método desenvolvido por R.S. Lashkari e K. R. Gunasingh (1990) procura resolver o problema de agrupamento de peças e máquinas em presença de planos de processos alternativos nos Sistemas Flexíveis de Manufatura. Um modelo de programação inteira 0-1, onde seqüencialmente primeiro agrupa as máquinas seguido da alocação das peças às máquinas. Um índice é desenvolvido para determinar a compatibilidade de uma peça produzida sobre um plano dado. Este índice pode ser definido de duas

maneiras: baseado nas exigências de ferramental ou no tempo de processamento $(S_{ipj})^2$.

$$(S_{ipj})^1 = T_{ipj} / \min(N_{ip}, M_j) \quad (4.1)$$

$$(S_{ipj})^2 = \frac{\sum_{z \in O_{ipj}} t_{iz}}{\sum_{r \in O_{ip}} t_{ir}} \quad (4.2)$$

onde:

- T_{ipj} = número de ferramentas comuns entre o plano "p" da peça "i" e a máquina "j";
- N_{ip} = número de ferramentas requeridas pela peça "i" usando plano "p";
- M_j = número de ferramentas disponíveis na máquina "j";
- t_{iz} = tempo de processamento da peça "i" usando a ferramenta "z";
- O_{ipj} = conjunto de operações na peça "i" sob o plano "p" usando a máquina do tipo "j";
- O_{ip} = conjunto de todas as operações a serem realizadas na peça "i" sob o plano "p".

O primeiro índice baseado em exigências de ferramental indica o movimento inter celular e o segundo índice o maior tempo de processamento de uma peça numa máquina, isto é, quanto maior a utilização da máquina se a peça e máquina são colocadas no mesmo grupo.

A função objetivo não linear é maximizar a soma dos índices de compatibilidade de todas as máquinas e peças, sobre todos os planos de processo em todos os grupos, isto é, comparar máquinas e peças em grupos comuns de maneira a ter o maior índice de compatibilidade possível. O modelo original foi decomposto em dois sub-modelos lineares, o qual é resolvido iterativamente até a solução final.

A função objetivo é do tipo:

$$\max Z_1 = \sum_i \sum_p \sum_j \sum_k X_{ipk} Y_{jk} S_{ipj} \quad (4.3)$$

onde:

$$X_{ipk} \begin{cases} = 1, & \text{se a peça "i" produzida sob o plano "p", está no grupo "k";} \\ = 0, & \text{em outro caso .} \end{cases} \quad (4.4)$$

$$Y_{jk} \begin{cases} = 1, & \text{se a máquina "j" está no grupo "k";} \\ = 0, & \text{em outro caso .} \end{cases} \quad (4.5)$$

As restrições do modelo são:

$$X_{ipk} = 1 \text{ para } \forall i \quad \begin{array}{l} \text{assegura que somente} \\ \text{um plano de processo} \\ \text{é selecionado para cada} \\ \text{peça e que uma peça} \\ \text{pertença somente a um grupo;} \end{array} \quad (4.6)$$

$$\sum_k Y_{jk} \leq A_j \text{ para } \forall j \quad \begin{array}{l} \text{restrições de utilização} \\ \text{do número de máquinas} \\ \text{e cada tipo no sistema;} \end{array} \quad (4.7)$$

$$\sum_i \sum_p X_{ipk} \leq H_k \text{ para } \forall k \quad \begin{array}{l} \text{restrições no número} \\ \text{de peças no grupo de} \\ \text{maneira a obter um} \\ \text{bom controle e supervisão} \\ \text{sobre os grupos individuais;} \end{array} \quad (4.8)$$

$$\sum_j Y_{jk} \leq G_k \text{ para } \forall k \quad \begin{array}{l} \text{restrições no número} \\ \text{de máquinas no grupo} \\ \text{de maneira a alcançar um} \\ \text{bom controle e supervisão} \\ \text{sobre os grupos individuais;} \end{array} \quad (4.9)$$

$$X_{ipj}, Y_{jk} \in \{0,1\} \text{ para } \forall (i,p,j,k) \quad \text{assegura integralidade} \quad (4.10)$$

das variáveis de decisão.

A divisão do modelo não linear em dois sub-modelos, um para alocação de máquinas e o outro para alocação de peças, é dada nas equações abaixo.

Para uma dada peça inicial \bar{X}_{ipk} alocada, o modelo de alocação de máquina (MA) é:

$$\max Z_2 = \sum_i \sum_p \sum_j \sum_k \bar{X}_{ipk} Y_{jk} S_{ipj} \quad (4.11)$$

sujeito à:

$$\sum_k Y_{jk} \leq A_j \text{ para } \forall j \quad (4.12)$$

$$\sum_j Y_{jk} \leq G_k \text{ para } \forall k \quad (4.13)$$

$$Y_{jk} \in \{0,1\} \text{ para } \forall j,k \quad (4.14)$$

sendo Y_{jk} a solução do problema de alocação de máquina (MA).

O correspondente sub-modelo de alocação de peças é formado como:

$$\max Z_3 = \sum_i \sum_p \sum_j \sum_k X_{ipk} \bar{Y}_{jk} S_{ipj} \quad (4.15)$$

sujeito à:

$$\sum_p \sum_k X_{ipk} = 1 \text{ para } \forall i \quad (4.16)$$

$$\sum_i \sum_p X_{ipk} \leq H_k \text{ para } \forall k \quad (4.17)$$

$$X_{ip} \in \{0,1\} \text{ para } \forall (i,k) \quad (4.18)$$

O modelo inicia-se com a alocação de uma peça variável na solução do agrupamento da máquina, e o resultado é usado para resolver o problema de agrupamento de máquina até que uma solução estável seja achada.

A metodologia foi baseada na existência de planos de processo múltiplos para cada peça. A divisão do modelo não linear em dois submodelos para otimização, pode não necessariamente alcançar o ótimo.

A função objetivo não leva em conta a capacidade da máquina ao minimizar custo. Uma dada peça inicial viável \bar{X}_{ipk} alocada é necessária para resolver o problema de alocação de máquina, e o sub modelo X_{ipk} é determinado subjetivamente e dependendo da solução inicial haverá modificações na solução final do modelo.

Outro problema é a não relação entre as peças de cada grupo e a capacidade de utilização da máquina, isto é, existe a possibilidade de alocar uma quantidade grande de peças em um grupo usando a mesma máquina, o que requer mais que a sua capacidade.

Usando ora o índice $S_{ipj}^{(1)}$, ora o $S_{ipj}^{(2)}$, nós não chegamos à mesma solução, porque os índices não são similares.

4.3 Conclusão

Todos os métodos apresentados para a solução de planos alternativos de processos utilizam o processo iterativo para a solução do problema, o qual pode não necessariamente alcançar o ótimo dependendo da solução inicial.

Existe uma série de modelos matemáticos para a solução do agrupamento de peças em presença de planos alternativos. Porém, as limitações dos métodos apresentados para a solução de tais modelos, nos levaram a presente proposta para a formação de família de peças baseado nos conceitos do Algoritmo Genético.

CAPÍTULO 5

5.- Modelo Proposto

Na avaliação da formação da célula de manufatura uma importante diferenciação em quase todos os métodos apresentados é a não incorporação de informações sobre a demanda. Esta variável afeta diretamente a utilização da máquina e as restrições de ferramental, que são partes significantes do custo total do sistema.

Algumas técnicas de formação de célula de manufatura apresentam alguns objetivos, como a minimização do movimento inter celular. Este procedimento não fornece para a célula um melhor desempenho global, pois a formação de uma célula de manufatura é uma tarefa de múltiplos objetivos.

Nesta proposta é assumido que uma peça pode ter mais que um plano de processo, e é mostrado como estes planos de processos alternativos influenciam na utilização dos recursos para a formação de família de peças e grupo de máquinas.

Um modelo de programação não linear inteira em 0 e 1 foi desenvolvido de maneira a formar o agrupamento de máquinas - peças. O modelo proposto não leva em consideração os movimentos intercelulares, ou seja, cada família de peças é alocada a um e somente um grupo.

5.1 Modelo Matemático

O desenvolvimento de um modelo de programação não linear inteira em 0 e 1 tenta simultaneamente identificar famílias de peças e respectivos grupos de máquinas. O modelo especifica os planos de processo de cada peça, os tipos de máquinas necessárias para executar cada peça baseado nas necessidades de ferramental das mesmas e o número de ferramentas disponíveis nas máquinas. Ele especifica o número total de máquinas requeridas para processar a peça, em função dos dados de produção como demanda, tempo e custos. Uma peça pode ser processada por um ou mais tipos de máquinas no grupo, ou uma máquina pode ser utilizada por mais de uma peça.

O modelo proposto foi formulado baseado nas seguintes premissas:

- a) informações sobre as necessidades de ferramental, assim como tempo e custo de processamento para cada peça usando as ferramentas necessárias e a quantidade de produção para cada peça;
- b) informações sobre o ferramental de cada máquina, custo da máquina e tempo de processamento disponível para cada máquina.

5.1.1 Notações

As seguintes notações foram utilizadas:

I - Índices

$i = 1, 2, \dots, n^o$, peças;
 $j = 1, 2, \dots, n^o$, máquinas;
 $k = 1, 2, \dots, n^o$, grupos;
 $p = 1, 2, \dots, n^o$, planos
 $l = 1, 2, \dots, n^o$, ferramentas.

II - Parâmetros

q_i = produção por unidade de tempo requerida da peça "i";
 C_j = custo fixo anual da máquina "j" devido a sua aquisição e manutenção;
 M_j = número de máquinas do tipo "j" disponível;
 P_k = número de peças no grupo "k";
 G_k = número de máquinas no grupo "k";
 t_{ipl} = tempo de operação da peça "i" usando a ferramenta "l" no plano "p";
 C_{ipl} = custo de operação da peça "i" usando a ferramenta "l" no plano "p";
 b_j = tempo disponível de cada máquina "j";
 R_{ip} = conjunto de ferramentas requeridas pela peça "i" no plano "p";
 $l \in O_{ipj}$ = conjunto de operações da peça "i" sobre o plano "p" que necessita de máquinas do tipo "j";
 U_k = limite superior de utilização da célula (percentagem)
 D_k = limite inferior de utilização da célula (percentagem)

III - Variáveis de decisão

$$X_{ipk} \begin{cases} = 1, & \text{se a peça "i" produzida sobre o plano "p" está no grupo "k";} \\ = 0, & \text{ao contrário;} \end{cases}$$

$$Y_{jk} \begin{cases} = 1, & \text{se a máquina "j" está no grupo "k"} \\ = 0, & \text{ao contrário;} \end{cases}$$

5.1.2 Modelo da Célula

O objetivo do modelo proposto é a formação simultânea de família de peças e grupo de máquinas de maneira a minimizar o custo total, o qual é definido como a soma do custo anual para processar todas as peças do sistema e o custo de amortização anual em investir na máquina "j".

A função objetivo para o problema é:

Minimizar,

$$Z = \sum_i \sum_p \sum_l \sum_k \sum_j C_{ipl} q_i X_{ipk} Y_{jk} + \sum_j \sum_k C_j Y_{jk} \quad (5.1)$$

Sujeito à:

$$\sum_p \sum_k X_{ipk} = 1; \quad \forall i \quad (5.2)$$

$$\sum_k Y_{jk} = M_j; \quad \forall j \quad (5.3)$$

$$\sum_i \sum_p X_{ipk} \leq P_k; \quad \forall k \quad (5.4)$$

$$\sum_j Y_{jk} \leq G_k; \quad \forall k \quad (5.5)$$

$$D_k \sum_j b_j \cdot Y_{jk} \leq \sum_i \sum_p \sum_l t_{ipl} q_i X_{ipk} \leq U_k \sum_j b_j \cdot Y_{jk}; \quad \forall k \quad (5.6)$$

$$X_{ipk}, Y_{jk} \in \{0,1\}; \quad \forall (i, p, k, j) \quad (5.7)$$

$$P_k, G_k \geq 0, \text{ e inteiro ; } \forall k \quad (5.8)$$

$$D_k, U_k \geq 0, \text{ e inteiro ; } \forall k \quad (5.9)$$

As restrições do modelo são dadas pelas equações (5.2) à (5.6). A restrição (5.2) garante que somente um plano de processo é selecionado para uma dada peça, e que a peça pertence somente a um grupo. A restrição (5.3) assegura a disponibilidade do número de máquinas de cada tipo no sistema. As restrições (5.4) e (5.5) garantem que o número de peças e de máquinas não são violados. A restrição (5.6) calcula a capacidade do grupo “k” em função do tempo de processamento das peças que pertencem ao grupo e ao mesmo tempo balanceia as máquinas no grupo “k”. As restrições (5.7) à (5.9) asseguram integralidade das variáveis de decisão.

A maioria dos métodos encontrados na literatura para a formação de células de manufatura não leva em consideração as restrições de ferramental. Geralmente num Sistema Flexível de Manufatura, as ferramentas para todas as operações que serão executadas por uma dada máquina deverão estar armazenadas no magazine. Logo, as ferramentas são recursos importantes num Sistema Flexível de Manufatura, uma vez que elas têm uma parcela significativa de participação no custo total do sistema. Além disto, é necessário estimar antecipadamente o número de cópias de cada tipo de ferramenta, de maneira a garantir que há uma oferta de ferramentas. O computador não pode enviar uma peça para ser processada por uma determinada máquina, se pelo menos todas as ferramentas requeridas para as próximas operações da peça não estiverem previamente colocadas (Stecke, K. El., 1983). Isto indica a necessidade do planejamento da produção. No problema de planejamento da produção, duas importantes considerações deverão ser feitas:

- a flexibilidade da folha de processo;
- as restrições de ferramental.

A flexibilidade da folha de processo é um problema de determinação da folha de processo para cada peça. No método proposto assume-se que uma peça pode ter mais que um plano de processo, contribuindo com isto para uma maior flexibilidade do modelo.

O Algoritmo Genético foi adotado pela sua poderosa habilidade de procura, sua robustez e sua adaptação fácil evidenciadas pelas suas múltiplas aplicações em diversos campos. No caso em estudo, conforme será visto neste

trabalho, o Algoritmo Genético se adapta à solução do problema de formação de células de manufatura de forma harmoniosa.

A seguir, veremos a aplicação do Algoritmo Genético ao modelo proposto.

5.2 Algoritmo Genético para a Otimização do Modelo Proposto

A utilização do Algoritmo Genético para a solução do problema de formação de células de manufatura representa uma poderosa ferramenta para alcançarmos os melhoramentos necessários no ciclo total de produção. Baseado no conceito de Tecnologia de Grupo e extraíndo os dados essenciais do planejamento de processo, o Algoritmo Genético será utilizado para determinar o melhor agrupamento.

Na aplicação do Algoritmo Genético, os seguintes passos devem ser seguidos:

- (i) Inicialização: inicializa pela criação de cromossomo representativo da população;
- (ii) Avaliação: avaliação em cada geração pela Função de Ajuste;
- (iii) Reprodução: baseado na avaliação pela Função de Ajuste uma nova população de cromossomo é selecionada para uma subseqüente operação de cruzamento e mutação;
- (iv) Cruzamento: acasalamento dos cromossomos de maneira a obter novos indivíduos;
- (v) Mutação: altera o valor de um gene de maneira a criar ocasionalmente, um novo cromossomo;
- (vi) Retornar: ao item (ii) até que um número pré definido de gerações seja alcançado.

O estabelecimento dos parâmetros apropriados do Algoritmo Genético é um processo de acertos (experimentos) e tentativas. Esta tarefa é tediosa e não existem regras formais. Todavia, o estabelecimento de parâmetros adequados pode melhorar o desempenho e a eficiência de forma bastante significativa. Os parâmetros do Algoritmo Genético necessitam ser especificados antes de cada experimento numérico, incluindo entre eles o valor da função ajuste (fitness), o número de gerações, o número de indivíduos em

cada população, probabilidade de cruzamento e de mutação. O principal propósito da implementação preliminar é achar a solução apropriada para o problema de formulação de célula de manufatura. Um número grande de experimentos usando combinações de parâmetros diferentes é mostrado no item 6.5.

5.2.1 Processo Lógico de Formação do Algoritmo Genético

Conforme visto anteriormente, o Algoritmo Genético é um processo que consiste em, pelo menos aproximadamente, copiar a maneira pela qual a vida vem se adaptando às condições ambientais do Planeta Terra. Este processo consiste em reproduzir elementos de uma população, cruzando-os entre si. Neste processo, a troca entre os pares de elementos, pode dar origem a indivíduos de formas diferentes. Porém, nem todos os novos elementos gerados possuem forma idêntica aos seus progenitores. Algumas diferenças serão mínimas e podem não afetar os descendentes, vez ou outra podem ocorrer diferenças fundamentais que podem colocar o processo em cheque. Os elementos que não se adaptarem não sobreviverão. Todavia pode ocorrer variações que sejam compatíveis com alterações no ambiente dando à população uma melhor adaptação ao meio. Assim, vemos que existe a necessidade de uma população, um código para passar informações, um processo de troca das informações e uma forma de testar a adaptabilidade da nova população.

No presente trabalho nos envolveremos com peças, máquinas, planos, ferramentas, custo, tempo de processamento, grupos de peças e grupos de máquinas. Esse conjunto de informações relaciona-se entre si, ou seja, as máquinas usarão ferramentas para produzirem peças, projetadas para serem executadas sob diferentes planos, e seus planos definem tempos, e custos de processamento diferentes. A tarefa que temos pela frente é então organizar esse conjunto de informações em uma seqüência lógica de dados para serem processados via código computacional. Ora organizar esse conjunto de dados em uma forma lógica é altamente complicado. Além do mais, esse código deve ter caráter geral e deve adaptar-se a uma gama muito ampla de problemas de formação de células de produção (manufatura). Embora venhamos a empregá-lo no presente trabalho para o problema de produção de peças mecânicas, isto não implica em sua aplicação somente a esta área.

O problema fundamental a ser resolvido no presente trabalho é minimizar o custo de fabricação de um conjunto de peças projetadas sob

diferentes planos usando um conjunto definido de máquinas. Sendo que peças e máquinas deverão ser agrupadas em células.

O método proposto para determinar o melhor arranjo dessas peças e máquinas em grupos é o método do Algoritmo Genético. Este método proposto por Holland, é um método de procura adaptativo, o qual se ajusta a cada domínio. Máquinas e peças formam um domínio de variáveis discretas, portanto o valor, custo mínimo de produção, é uma função também discreta. Uma das grandes vantagens da aplicação do Algoritmo Genético é a sua adaptabilidade a esse tipo de problema (Goldberg D. E.,1989). O algoritmo requer que as variáveis de seu domínio sejam bem conhecidas, ou seja, as peças, ferramentas, planos, etc... têm que ser especificados a priori e sem possibilidade de troca, isto é, a solução será válida para aquele determinado domínio. Estas informações sobre as variáveis pertencentes ao domínio formam o conjunto de dados que temos para criar o código genético, ou seja, o conteúdo que deverá ser trocado no processo de criação(genes) das futuras populações. É bom observar que as informações iniciais permanecem, o que muda são as formas como serão combinadas entre si. A combinação destas informações em uma seqüência forma uma cadeia lógica de instruções. A essa cadeia chamamos de cromossomo (muito semelhante ao DNA) e a cada informação, conforme sua posição na cadeia, um alelo.

5.2.2 Criação do Código na Cadeia

Conforme visto, as informações iniciais devem ser combinadas de forma tal que ao criarmos um código a ser transmitido de indivíduo a indivíduo, ao longo do processo de reprodução, estas informações não sejam destruídas ou modificadas. Devem a cada instante poder identificar peças, ferramentas, etc...

No método do Algoritmo Genético costuma usar-se os dígitos "0" e "1". Todavia, devido à quantidade de informações a serem transmitidas e identificadas, isto geraria uma cadeia muito longa . Claro que poderíamos ter optado por este sistema mas, devido a características do problema que temos em mãos, foi possível a criação de um sistema de codificação mais simples .

Este sistema de codificação, o qual passaremos a descrever, aproveita a associação entre peças, planos, ferramentas e máquinas, etc... para poder interrelacioná-las de tal forma a criar o cromossomo e seus alelos.

Cada cromossomo tem seu comprimento, ou seja, um número de alelos igual ao número de peças. Podemos considerar pictoricamente um cromossomo como uma seqüência de casas. Mostramos na figura 5.1 a construção fundamental, ou seja, devemos ter o número de casas igual ao

número de peças. Assim, cada casa corresponde a uma peça. As casas estão inicialmente vazias. Podemos, então, associar cada peça a um grupo. O número de grupos, sempre mais do que um, é determinado ao longo do processo de minimização.

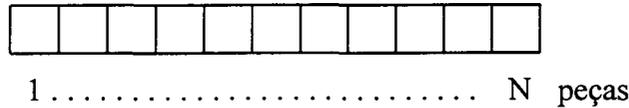


Figura 5.1: Representação de um cromossomo

Assim mostramos na figura 5.2 uma série de números associados às casas. As casas que possuem o mesmo dígito pertencem ao mesmo grupo, ou seja, as peças (3, 4,...) pertencem ao grupo "1", as peças (2,...,N-2, N-1, N) pertencem ao grupo 3 e as peças (1,...,N-4, N-3) pertencem ao grupo 2. Temos, neste exemplo pelo menos três grupos.

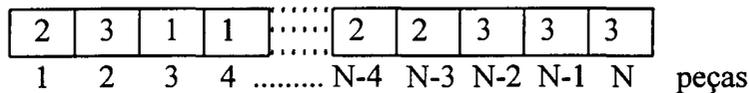


Figura 5.2: Representação do cromossomo peça-grupo

Agora devemos criar um código que informe ao cromossomo qual o plano de fabricação da peça. É importante salientar que o plano de fabricação da peça carrega consigo as ferramentas a serem usadas, bem como seu tempo e custo de fabricação. Ora cada peça pode ser feita de inúmeras formas diferentes. Assim, o número de planos por peça é um dado de entrada e depende das peças e máquinas disponíveis. A forma encontrada, e aqui proposta é de associar ao cromossomo fundamental uma outra seqüência de casas vazias, sobrepostas as casas do grupos. Na figura 5.3 mostramos uma seqüência de casas de grupos e o número de planos por peças.

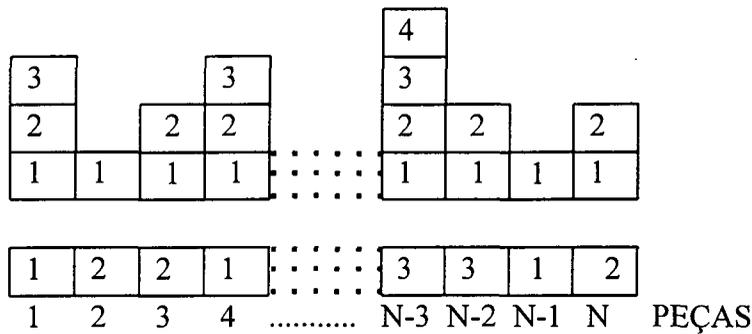


Figura 5.3: Representação de um cromossomo peça-grupo-plano

Na figura 5.3 mostramos que a peça N-3, que pertence ao grupo 3, pode ser fabricada de quatro formas distintas, porém o cromossomo só pode carregar consigo um plano por alelo. Para contornar este problema foi proposta a alocação de uma fileira de N casas vazias adicionais, superpostas às casas de grupos, formando o cromossomo de cadeias vazias mostradas na figura 5.3.

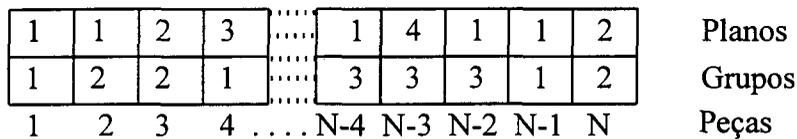


Figura 5.4: Exemplo de um cromossomo peça-grupo-plano

Vemos, portanto na figura 5.4, que, por exemplo, a peça N-3 pertence ao grupo 3 e será feita sob o plano 4. Esta representação codificada de peças, planos, grupos e ferramentas forma o esquema da primeira parte do cromossomo. A segunda parte do cromossomo está associada às máquinas disponíveis para a fabricação das peças. Ora, cada peça tendo seu plano de produção deverá usar uma ou várias máquinas disponíveis. No código aqui proposto, devemos ter no mínimo uma máquina disponível de cada tipo requerida pelos planos de fabricação. Esta particularidade abre a possibilidade de existirem várias máquinas de um mesmo tipo. Para codificar estas informações procedeu-se de forma semelhante ao que foi proposto para o conjunto de peças. Desta forma a segunda parte do cromossomo terá uma cadeia de casas e de comprimento igual ao número de máquinas, conforme visto na figura 5.5.

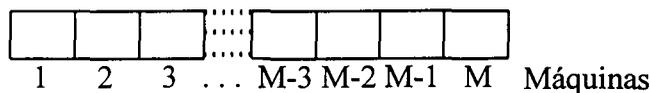


Figura 5.5: Representação do cromossomo máquina

As casas estão vazias, e para preenchê-las devemos verificar a que grupo elas pertencem. Assim o numeral colocado dentro da casa vazia da máquina informará a qual grupo ela pertence. Mostramos na figura 5.6 uma seqüência parcial do cromossomo máquina, com as casas preenchidas pelo número do grupo a que cada máquina pertence. Por exemplo, a máquina 3 pertence ao grupo 2(dois) e as máquinas (1,2,...,M) pertencem ao grupo 1(um).



Figura 5.6: Representação do cromossomo máquina-grupo

Uma propriedade importante do código aqui proposto é que podemos associar às "M" casas um número ilimitado de máquinas de um mesmo tipo. Isto é feito alocando a cada posição tantas casas vazias quantas forem as máquinas disponíveis daquele tipo. Sobrepondo então à cadeia mostrada na figura 5.6, as casas vazias correspondentes às máquinas disponíveis de cada tipo, teremos a disposição codificada mostrada na figura 5.7.

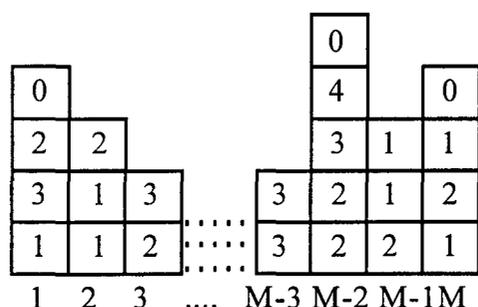


Figura 5.7: Exemplo de um cromossomo máquina-grupo

A figura 5.7 é uma representação mais real da situação do grupo de máquinas disponíveis para a fabricação das "N" peças, ou seja, tomemos por exemplo a máquina (M-2). São disponíveis para o uso 5(cinco) máquinas deste tipo, porque temos 5(cinco) casas a serem alocadas a grupos. No exemplo mostrado na figura 5.7, os grupos 2, 3 e 4 deverão usá-la, sendo que no grupo 2 existe a necessidade de duas máquinas do tipo (M-2). O dígito 0(zero) indica que a quinta máquina do tipo (M-2) estará ociosa, isto é, não será usada por qualquer grupo. Para sabermos como será formado um grupo de máquinas,

$$\begin{array}{rcc}
 \begin{array}{c} (A) \\ \backslash A \end{array} + \begin{array}{c} (C) \\ \backslash C \end{array} = \begin{array}{c} AC \\ AC \\ AC \\ AC \end{array} \left. \vphantom{\begin{array}{c} (A) \\ \backslash A \end{array}} \right\} \text{castanho} \\
 \text{puro} \quad \text{puro} \\
 \\
 \begin{array}{c} (A) \\ \backslash A \end{array} + \begin{array}{c} (A) \\ \backslash C \end{array} = \begin{array}{c} AA \\ AC \\ AA \\ AC \end{array} \left. \vphantom{\begin{array}{c} (A) \\ \backslash A \end{array}} \right\} \begin{array}{l} - \text{azul} \\ - \text{castanho} \\ - \text{azul} \\ - \text{castanho} \end{array} \\
 \text{puro} \quad \text{híbrido} \\
 \\
 \begin{array}{c} (A) \\ \backslash C \end{array} + \begin{array}{c} (A) \\ \backslash C \end{array} = \begin{array}{c} AA \\ AC \\ AA \\ AC \end{array} \left. \vphantom{\begin{array}{c} (A) \\ \backslash C \end{array}} \right\} \begin{array}{l} - \text{azul} \\ - \text{castanho} \\ - \text{castanho} \\ - \text{castanho} \end{array} \\
 \text{híbrido} \quad \text{híbrido}
 \end{array}$$

Figura 5.9: Exemplo de cruzamento de Alelo

A esse mecanismo chamamos de dominância. Este mecanismo, mesmo em Engenharia Genética, é assunto de intensa atividade de pesquisa. Como não é de nossa área a discussão deste problema, mas sim criar no código esta propriedade, vamos tentar reproduzi-lo da melhor forma possível. Para tal fim associado a cada cromossomo do tipo completo, mostrado na figura 5.9, será criado o seu homólogo. Logo, no mecanismo de reprodução teremos sempre presente um par de cromossomos, um recessivo e um dominante, sendo que o cromossomo dominante, aquele que aparentemente é o que dá a forma ao indivíduo, será chamado de cromossomo efetivo.

Na figura 5.10 mostramos esquematicamente o conjunto destes cromossomos.

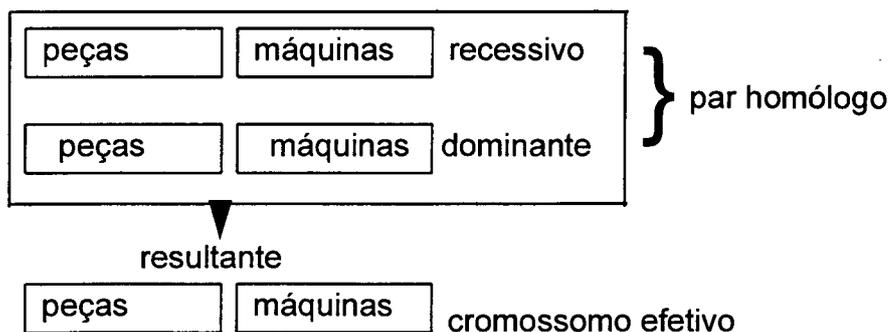


Figura 5.10: Representação do cromossomo efetivo

Na figura 5.10 procura mostrar-se que no processo de reprodução para criar uma nova população, um cromossomo efetivo é aquele que sempre dá as características. Porém, junto com ele segue sempre o cromossomo recessivo, formando um par homólogo. No cruzamento participam então, um conjunto de quatro cromossomos formando dois pares homólogos. Resta-nos, portanto, agora, criarmos uma informação que possibilite a identificação do cromossomo com características de dominância. Foi visto que cada peça está associada a um plano e que a cada gene do cromossomo corresponde um grupo. Se cada peça corresponde a um grupo, teremos tantos grupos quanto for o número de peças que é uma solução trivial do problema de formação de células. Todavia, todas as peças formando um só grupo é praticamente impossível. De uma forma ou de outra, o número ideal de grupos é a incógnita e deve ser determinado ao longo do processo de otimização. A reprodução inicia-se sempre considerando que o número de grupos é igual ao número de peças, isto é, teremos tantos grupos quantas forem as peças. Como esta é a solução trivial, qualquer outro cromossomo que apresente um número menor de grupos terá mais chance de se adaptar

(ter melhor valor da função ajuste). Devido a sua maior adaptabilidade, este cromossomo, por ter maiores chances, dominará sobre seu par homólogo. Forma-se assim uma Heurística, que ajuda no agrupamento das peças e máquinas, e, como tal, funcionando como um problema de multi-objetivo.

Seja o par mostrado na figura 5.11 na qual mostramos as peças e os grupos.

1	3	3	2	2	1	1	3	1	4	4	1	MÁQ	(1)
1	2	3	4	5	6	7	8	9	10	11	12		
Peças													

1	1	2	3	3	3	2	1	2	1	2	2	MÁQ	(2)
1	2	3	4	5	6	7	8	9	10	11	12		
Peças													

Figura 5.11: Exemplo de um par de cromossomo

No cromossomo "1" contamos quatro grupos, ou seja, o grupo "1" (um) é formado pelas peças (1,6,7,9,12); o grupo "2" (dois), que é formado pelas peças (4,5); o grupo "3" (três), do qual constam as peças (2,3,8); e finalmente o grupo "4" (quatro), no qual temos as peças (10,11). Já o cromossomo "2" do par homólogo é formado pelos grupos : "1" (um) do qual

constam as peças(1,2,8,10); o grupo "2" (dois), no qual pertencem as peças (3,7,9,11,12); e por último, o grupo "3" (três), do qual fazem parte as peças (4,5,6). Como o cromossomo "2" do par homólogo tem três grupos de peças, ele será o dominante, pois tem um número menor de grupos, será portanto o efetivo. Ao participar, portanto, num cruzamento para formar um novo indivíduo, o efetivo será sempre aquele com o menor número de grupos de peças. Criamos assim uma seqüência lógica de codificação e controle dos cromossomos e de seus genes, resta-nos ver agora como esse processo pode, geração após geração, evoluir e fornecer uma melhoria geral na espécie (no caso em questão uma solução otimizada do problema).

5.2.3 Criar uma população

Ora, criar um código, conforme vimos, é uma tarefa complexa, além disto devemos ainda estabelecer critérios para que este código não perca a sua identidade, geração após geração. O processo tem seu início no ato de criação da primeira população. O tamanho da população é determinado aleatoriamente e é naturalmente um número inteiro, cabe, portanto, ao usuário a sua escolha. Já o comprimento total do cromossomo é igual a soma do número de peças com o número de máquinas que está diretamente ligado à natureza do problema em estudo. Outro valor livremente determinável é o número máximo de máquinas por grupo. No processo de evolução natural, o futuro é indeterminado e não sabemos o objetivo a atingir, já no nosso problema o objetivo é claro e é controlado por uma função de ajuste, que geração após geração, deve ser otimizada. Desta forma, ao longo do processo de reprodução, cada geração nova deverá estar melhor adaptada à função ajuste e caso encontremos a melhor solução o processo parará e a solução ótima será encontrada e apresentada. Necessário se faz, porém, antecipadamente especificar o número máximo de gerações.

A formação da primeira geração é feita por sorteio, isto é, cada casa do cromossomo é sorteada, utilizando uma espécie de dado, que possui inicialmente um número de faces igual ao número de grupos. Também os planos são sorteados e cada dado terá um número de faces igual ao número de planos. Na parte do cromossomo onde encontram-se as máquinas, o procedimento é idêntico. Isto também é feito para o homólogo, logo, em uma população de " N " cromossomos, teremos, na realidade, o dobro dos cromossomos, o dominante e o recessivo. Criada desta forma a população inicial, dá-se então seqüência ao processo de reprodução. O processo de reprodução é feito casando-se (cruzando) entre si cromossomos da população

inicial. Como cada cromossomo da população é diplóide, estarão, na realidade, quatro cromossomos trocando informações entre si.

Na figura 5.12, o cromossomo "AB", o dominante e o recessivo do elemento 1 da população original, cruza com o elemento N da mesma população. Deste cruzamento geramos um novo par de cromossomos "1" da nova população e um novo elemento "N". Esta diferença é mostrada graficamente na forma dos desenhos internos da figura 5.12. Assim, é fácil verificar que o número de elementos da população permanece o mesmo desde o início do processo reprodutivo. Aqui é importante perguntar, ou melhor, fazer uma indagação de como um cromossomo escolhe um par para fazer o cruzamento. Para que o processo tenha pelo menos estatisticamente uma chance de caminhar em direção à melhor solução otimizada, é necessário que os elementos mais aptos sobrevivam e os menos aptos morram. Assim, para cada cromossomo efetivo é calculada a função ajuste, e àqueles cromossomos com melhor valor desta função, é associada uma maior probabilidade de ser escolhido para cruzamento. Agindo desta forma, podemos garantir que a nova população será formada de indivíduos com um maior valor da função ajuste, inclusive na média. Vimos que para criar uma nova população, a partir de uma população inicial gerada aleatoriamente, é necessário um mecanismo de troca de informações. Esta troca é feita pelo acasalamento dos elementos da população inicial dando aos elementos mais ajustados uma melhor chance de sobreviverem. Na verdade, o acasalamento corresponde a uma troca de genes. Para a efetiva troca de genes entre dois cromossomos é necessário o estabelecimento de algumas regras simples. A primeira forma é a troca de partes de cada cromossomo entre si. Na figura 5.12 o cromossomo "1" e o "N" trocam entre si partes das suas cadeias dando origem a dois novos indivíduos totalmente diferentes.

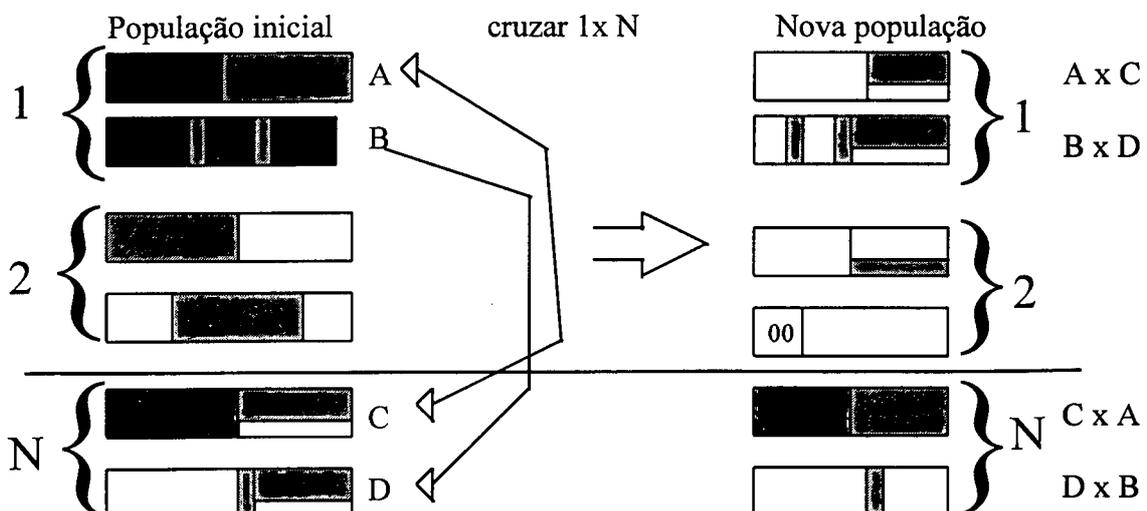


Figura 5.12: Criação da Nova População

Porém, neste processo os cromossomos trocam entre si partes inteiras cruzando então as informações. A esta operação damos o nome de cruzamento (crossover). Nela podemos verificar que partes do cromossomo "1A" passam para o cromossomo "NC" e formam o cromossomo "AC" e "BD" da nova população. Este procedimento cria uma população inteiramente nova, sendo que este processo é repetido até que o valor otimizado seja encontrado. Existem vários métodos na literatura para proceder o cruzamento (Goldberg, 1989), sendo que no presente trabalho foram utilizados os métodos PMX e OX.

No problema da reprodução usando o cruzamento para a criação da nova população, parte dos cromossomos são trocados entre si. Assim, na verdade, os indivíduos não são capazes de por si só criarem novos elementos totalmente diversos de seus progenitores, mas eles estão só recombinao a informação original. O processo de cruzamento pode levar a população em uma determinada linha evolutiva e que muitas vezes não é a direção desejada, e às vezes torna-se totalmente impossível reverter este rumo. Para contornar esta dificuldade, eventualmente algum elemento da população sofre a alteração de um de seus genes de forma randômica. Na natureza o processo evolucionário é lento e as mutações são raras.

A natureza do problema é que determina a probabilidade de ocorrência de uma mutação, sendo que na maioria dos problemas esta probabilidade é muito pequena. No problema em questão e que está sobre estudo, a função objetivo é penalizada quando o valor calculado não tem sentido real. Quando ocorrer uma mutação neste valor, ela não é passada à geração seguinte, razão porque, para esta situação, é necessário que a probabilidade de mutação seja relativamente alta.

5.3 Considerações sob a Função Ajuste

O modelo em pauta é na realidade um problema de minimização. Como o Algoritmo Genético foi desenvolvido para determinar o máximo de uma função vamos proceder da seguinte forma. Seja uma função $f(x)$ que possua um mínimo, conforme mostramos na figura 5.13 .

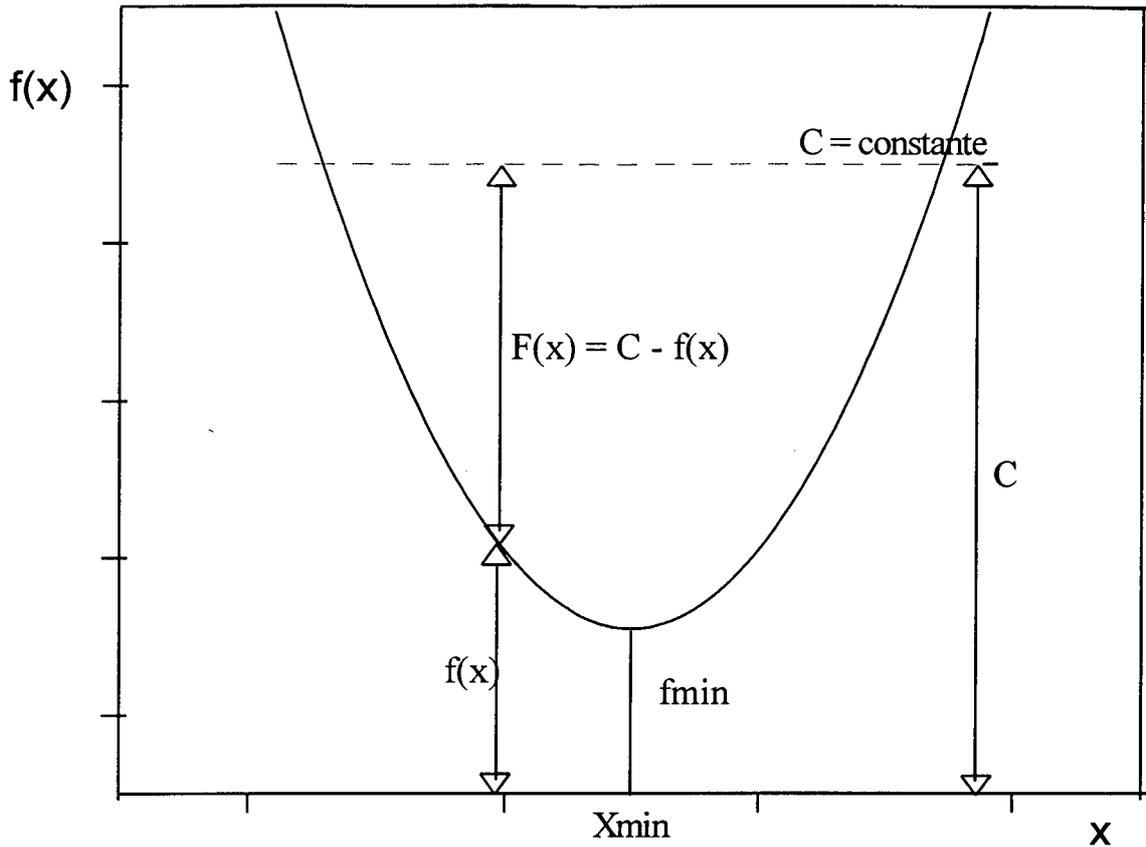


Figura 5.13: Representação de uma função com mínimo

Na figura 5.13, o valor da função $f(x)$ no ponto X_{\min} é o mínimo procurado. Todavia podemos construir a função

$$F(x) = C - f(x) \quad (5.10)$$

onde C é uma constante grande o suficiente tal que o $\max F(x) \Rightarrow \min f(x)$, ou seja, como pode ser visto claramente na figura 5.14, no ponto X_{\max} a função $F(x)$ tem um máximo.

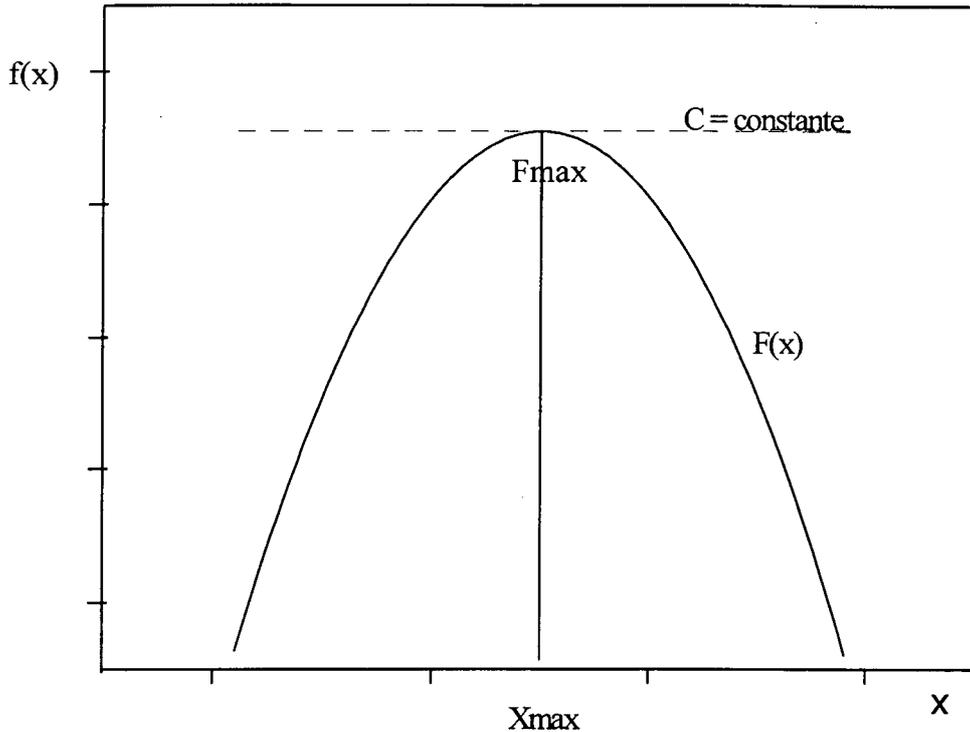


Figura 5.14: Representação da Função Ajuste

Ora, de uma maneira genérica estamos então determinando o máximo da função objetivo (aqui representada genericamente por $F(x)$). Procuramos ilustrar neste exemplo, por intermédio de uma função bidimensional, o comportamento do problema frente a uma equação (função) simples. No problema que estamos estudando, estamos na realidade com uma função de várias variáveis. Mas o problema mesmo em espaço n -dimensional pode ser atacado da mesma forma. Ou seja, determinamos o maior valor da Função Ajuste na população inicial, fazemos a constante C na equação 5.10 igual ao valor máximo e continuamos o processo, caso dê negativo, isto é, se existir um valor da $f(x)$ tal que $F(x) < 0$, multiplicamos então o valor da $f(x)$ por um número inteiro, 10 por exemplo, e continuamos a operação. Procedendo-se desta forma este problema é contornado, sendo que o Algoritmo absorve bem este tipo de solução sem provocar qualquer tipo de perturbação no método aqui desenvolvido.

Por exemplo, seja a função

$$f(x) = x^2$$

cujos mínimos são $f(x) = 0$ para $x = 0$ e o gráfico correspondente à figura 5.13 é mostrado na figura 5.15.

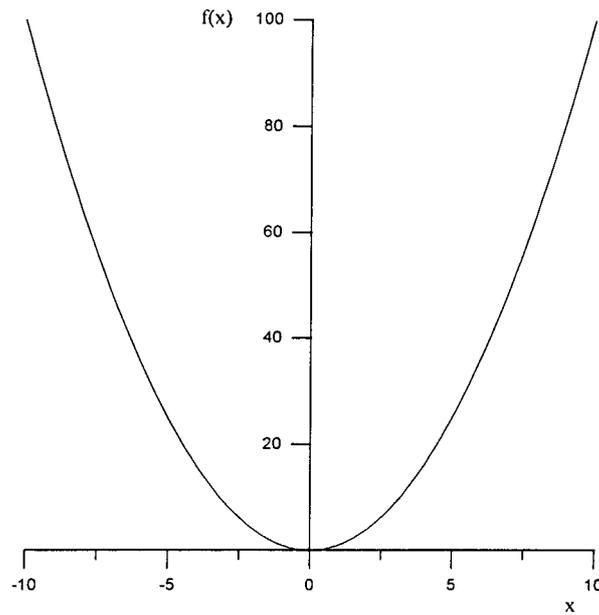


Figura 5.15: Representação da Função $f(x) = x^2$

Fazendo $C = 10$ e substituindo este valor na função $F(x) = C - x^2$, teremos a função:

$$F(x) = 10 - x^2$$

cujo máximo será $F(x) = 10$ para $x = 0$, como mostrado na figura 5.16

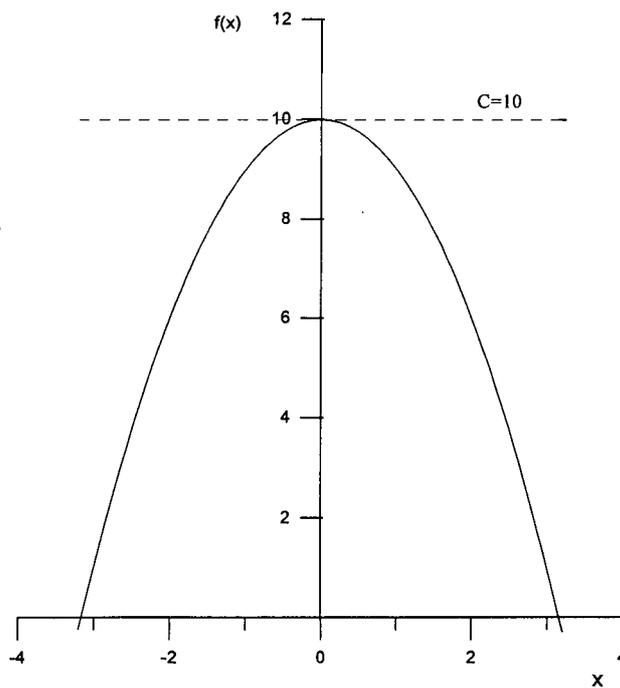


Figura 5.16: Representação da Função $F(x) = C - x^2$

CAPÍTULO 6

6. - Implantação do Modelo

6.1 Considerações Gerais

Conforme visto no capítulo anterior, o Algoritmo Genético é um método de otimização que usa um mecanismo de procura. A idéia fundamental deste algoritmo é imitar os procedimentos desenvolvidos na natureza para manter a vida no nosso planeta. Nas discussões anteriores foi mostrado como estes procedimentos podem ser representados algebricamente. Ora, todo procedimento possível de ser representado matematicamente se enquadra com a idéia de ordem e lógica, sendo portanto programável. A ordenação das operações e procedimentos adotados no Algoritmo Genético foram, após exaustivo estudo, analisados e programados num código computacional. Como o algoritmo vem sendo pesquisado por uma série de outros autores, conforme Davis (1991), existe na prática desta área de pesquisa outros códigos computacionais. Desta forma, alguns itens do código aqui proposto tem estreita semelhança com alguns deles, principalmente com o de Venugopal (Davis, 1991).

Conforme já mencionado, o Algoritmo Genético é uma combinação entre a sobrevivência dos mais aptos e trocas de informações aleatórias, com a finalidade de criar um método de procura com um leve toque humano de análise. Novas gerações são criadas usando partes das mais antigas e mais aptas. O ponto central do Algoritmo Genético é, portanto, um compromisso entre eficiência e eficácia com o objetivo de apoiar a sobrevivência dos descendentes mais aptos. Esse princípio deve ser uma qualidade básica para qualquer espécie de sistema que se desenvolva seguindo esta linha de raciocínio. Ao se construir sistemas computacionais artificiais tentando imitar sistemas biológicos, devemos procurar atingir alto grau de adaptação e flexibilidade, o que, conforme visto, não é uma propriedade dos métodos tradicionais. Quanto mais se estuda e se trabalha com os sistemas biológicos, verifica-se cada vez mais a sua alta eficiência e flexibilidade. Qualidades como auto-correção e auto-orientação são inerentes aos sistemas biológicos, enquanto em sistemas artificiais elas são difíceis de serem criadas.

Nesta explanação inicial de sistemas computacionais otimizantes chega-se à conclusão de que a simulação artificial por imitação de processos

biológicos representa uma área de atividade promissora. O segredo consiste em estudar-se com muita atenção e cuidado a forma como os sistemas naturais se adaptam, se reproduzem e evoluem em sistemas tão adversos. O simples fato do Algoritmo Genético ser uma imitação muito pobre de como os cromossomos, no processo de evolução das espécies, sobrevivem até hoje nos diversos ambientes terrestres, não é o único argumento para aceitá-lo. Durante o seu desenvolvimento desde o trabalho pioneiro de Holland (1971), o Algoritmo Genético provou através de muitas aplicações, em situações complexas, a sua robustez. A razão é muito clara, este Algoritmo é simples de ser programado, principalmente orientado para o objeto, e sua potencialidade é muito elevada, visto a sua total independência às restrições de unicidade, continuidade e derivabilidade.

6.2 Código Computacional

No item anterior foi colocado que o Algoritmo Genético possui uma série de propriedades que o classificam, de certa forma, numa escala evolucionária, como mais moderno e mais orientado à programação. Aceitar como dogma esta supremacia do Algoritmo Genético sobre os métodos tradicionais é ainda prematuro. De uma forma geral o Algoritmo Genético é diferente dos outros métodos de otimização nos seguintes aspectos:

I - O Algoritmo Genético funciona baseado numa codificação do conjunto de variáveis e não com as variáveis físicas do problema

No presente trabalho as variáveis do problema são números de peças (na realidade tipos de peças), número de máquinas (na realidade tipos de máquinas), planos de execução de cada peça, usando os diferentes tipos de máquinas (na realidade operações de fabricação), custos e tempo. Conforme visto no capítulo 5, estas variáveis foram codificadas. A partir da codificação, o problema torna-se abstrato e o procedimento passa a ser executado a nível de código. Nos métodos tradicionais trabalha-se com as próprias variáveis em estudo.

II - O Algoritmo Genético trabalha com uma população de variáveis codificadas do problema e não com um único valor da função qualquer que seja sua forma.

Um dos parâmetros mais importantes do Algoritmo Genético é o tamanho da população. Ele é de fundamental importância para o sucesso e velocidade do método. As circunstâncias que levam à evolução nas gerações subsequentes estão intimamente ligadas a esse número, segundo a teoria de equilíbrio pontuada por Gould, S.J. (1970), desenvolvida na década de 70 para o sistema biológico terrestre. Resumidamente, a evolução ocorre em pequenos nichos biológicos. É necessário um evento incomum na população permitindo a separação de alguns genes de um elemento da população que provoquem alguma melhoria sensível, sem que provoquem efeitos colaterais ruins. Como, por exemplo, suponhamos que uma alteração em um gene cria um dente nas aves em vez de bicos, porém um gene de um cromossomo não o muda por completo e a alteração de um único gene é difícil de ocorrer. Junto a essa troca poderia ocorrer a alteração de um elemento colateral aleatório, o que poria em risco a espécie. Desta forma é necessário uma pequena população isolada e protegida em algum lugar durante várias gerações. Esta situação pode favorecer o aparecimento de dentes ou dizimar aquela população isolada. Ora em pontos isolados teremos o aparecimento de novas espécies. De fato, ao analisarmos os resultados obtidos neste trabalho, veremos que cada vez que realizarmos um experimento no computador trabalhamos com uma população isolada, havendo casos que nenhum ótimo é obtido, considerando-se assim este experimento como degenerado (significando que não ocorreu evolução na população). Esta idéia de equilíbrio pontuado adapta-se bem ao espírito deste trabalho. Cada vez que rodamos um caso no computador, estamos na verdade realizando uma experiência genética. Nada nos garante que o resultado final desejado será atingido. Portanto, é necessário um estudo estatístico sobre o desempenho do método, quanto a seu comportamento frente a variações de seus parâmetros. Para fins de análise dos resultados, chamamos a este tipo de estudo "análise paramétrica dos atributos do Algoritmo Genético". Esta análise servirá para fixar o intervalo de variabilidade dos diversos parâmetros do Algoritmo Genético

III - O Algoritmo Genético controla seu desempenho, ou melhor, a adaptabilidade em relação a seus objetivos, por intermédio de uma função a qual é calculada a cada população para todos os seus indivíduos, não necessitando de derivadas, gradientes ou quaisquer outros operadores matemáticos.

A construção da Função Ajuste (função objetivo) é uma operação fundamental no Algoritmo Genético.

No processo de seleção natural, as condições do meio ambiente definem a adaptabilidade dos indivíduos. É ela que define os que vão sobreviver e os que vão morrer. Em outras palavras, ela é uma função controle. Cada problema a ser resolvido pelo Algoritmo Genético deve de algum modo possuir estas características, ou então não será possível resolvê-lo por este método. O código criado para representar as variações do problema deve adaptar-se a esta Função Ajuste, isto é, a cada alteração das variáveis deve ser possível a avaliação do valor da função, comparando este resultado entre os elementos da população.

IV - O Algoritmo Genético relaciona, cria, modifica e compara suas diversas variáveis entre si de forma probabilística e não determinística.

No Algoritmo Genético, geração após geração, a população vai sendo modificada. Neste processo de modificação da população seus elementos vão cruzando entre si informações. No processo de troca de informações entre seus elementos são usados dois mecanismos fundamentais, o cruzamento e a mutação de algumas delas. O mecanismo de cruzamento de informações consiste na troca de partes da codificação das variáveis entre si. Isto gera dois indivíduos completamente diferentes. A função Ajuste, conforme definida no requisito III, avalia estes elementos quanto a melhor adaptabilidade e elimina probabilisticamente o menos apto. Para que ocorra cruzamento há a necessidade de definir-se a probabilidade de ocorrência de cruzamento, conforme definida no capítulo 3. Para um par de indivíduos simples (sem o uso de conceito de gabaritos, visto no capítulo 3), a probabilidade de cruzamento é igual a "1", porém neste trabalho utiliza-se o conceito de gabarito, e esta probabilidade

indica as suas possibilidades de sobrevivência. Então o efeito combinado da reprodução com a probabilidade de cruzamento e sobrevivência dos gabaritos mais aptos funciona como um atenuador no crescimento geométrico do número de informações trocadas, dando assim mais estabilidade ao processo. A essa altura, a probabilidade de cruzamento surge como outro parâmetro importante do método. Uma análise do efeito de sua variabilidade no resultado do algoritmo faz parte do estado paramétrico anteriormente citado. O mecanismo de mutação consiste na alteração aleatória de informações, correspondendo no código a uma modificação aleatória de um dos seus elementos. Como estas alterações processam-se de forma totalmente inesperada e em local imprevisto, bem como em uma geração qualquer, há a possibilidade dela ser totalmente destrutiva. Todavia ela é necessária, pois pode alterar o direcionamento de todo o processo e orientá-lo em direção totalmente diversa. Como ela não ocorre em todos os elementos da população, pode ser assimilada e reprocessada, e de forma totalmente arbitrária e imprevista ser incorporada em algum cromossomo, seja ele dominante ou recessivo. Mais tarde, em outras gerações, pode provocar uma alteração qualquer. Testes devem ser produzidos para verificar a correlação existente entre o número de mutações, o tamanho da população, o número de gerações e a probabilidade de cruzamento. No processo natural as mutações são raras, porém não surtem o efeito tão destrutivo em população pequena e isolada.

Na discussão precedente foi examinado o algoritmo no seu aspecto mais genérico, isto é, foi visto que quatro são os parâmetros fundamentais do Algoritmo Genético, quais sejam: a) o número de elementos que compõem a população; b) o número de gerações; c) a probabilidade de cruzamento e; d) a probabilidade de mutação. Estes parâmetros são fundamentais para o Algoritmo Genético e não têm qualquer relação com o problema a ser otimizado.

Para a estruturação do programa que operará o Algoritmo Genético, os passos fundamentais são:

1 - Geração dos parâmetros fundamentais

→ Tamanho da população	Pop_size
→ Número de gerações	Num_ger
→ Probabilidade de cruzamento	Prob_cruz
→ Probabilidade de mutação	Prob_mut

2 - Inicialização do problema

- Criação do código
- Dados do problema
- Criação da população inicial de forma randômica conforme os parâmetros fundamentais do Algoritmo Genético

3 - Formulação da Função de Ajuste - Função Objetivo

→ Neste ponto, a Função Ajuste faz uma avaliação de todos os cromossomos da população e classifica-os em mais ou menos adaptados.

4 - Inicialização da reprodução

→ Fundamentado na classificação da população anterior, os elementos são selecionados para participarem nas operações subseqüentes de cruzamentos e mutações. Todos os elementos que foram classificados como mais adaptados terão maior chance de participarem na criação da população subseqüente.

5 - Cruzamento

→ Um programa para processar os cruzamentos é a parte fundamental desta fase do Algoritmo. Os pares são selecionados através de sorteio entre aqueles elementos que possuem mais adaptabilidade.

6 - Mutação

→ Ocasionalmente alguns elementos da população são sorteados e um novo elemento é criado aleatoriamente por alteração de um de seus genes.

7 - Processo Iterativo

→ O processo iterativo consiste em retornar ao passo 3 para processar a nova geração, avaliando e recriando até que o valor ótimo seja alcançado. Porém, como não se sabe a priori qual será esse valor, o procedimento normal é limitar o número de gerações e depois proceder uma análise do resultado alcançado, selecionando o valor ótimo encontrado para aquele experimento.

O código computacional, aqui proposto, incorpora estes sete passos em seqüência.. O código denominado daqui para a frente como "*GACELL*" (Genetic Algorithm for Cell Formation) foi preparado para processar e resolver problemas de formação de células de manufatura envolvendo custos e tempos de fabricação. A descrição detalhada do algoritmo é por demais longa e tediosa. Não será feita no texto deste trabalho uma análise passo a passo do programa.

Como o objetivo do presente trabalho é aplicar o Algoritmo Genético ao problema de formação de Célula de Manufatura, a linguagem em que foi escrito o código computacional é irrelevante para a validação dos procedimentos.

6.3 Estrutura do Sistema

Um diagrama geral do fluxo de dados do sistema para a implementação computacional do algoritmo para a formação de células de manufatura é resumido na figura 6.1.

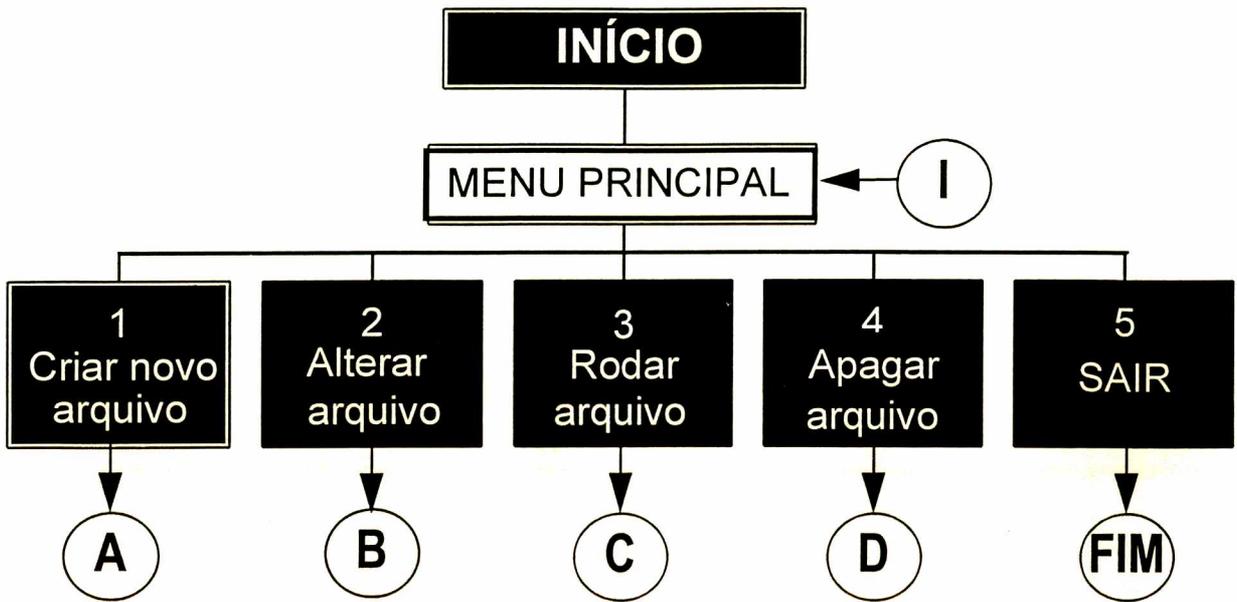


Figura 6.1 : Fluxo de Dados do Sistema

Neste menu principal, o usuário pode criar um novo arquivo, alterar um arquivo já existente, rodá-lo, apagá-lo e sair do sistema.

Para facilitar o manuseio do código *GACELL* pelo usuário foram criados programas conversacionais em que são feitas uma série de perguntas, que ao serem respondidas pelo operador, desde que corretas, dão seqüência automática aos subseqüentes passos, até a inicialização dos cálculos. O operador pode acompanhar via tela o desenvolvimento das gerações até o final, onde ele pode imprimir relatórios em forma de tabela ou gráficos.

O anexo I mostra a primeira janela exposta na tela, na qual aparecem cinco opções, para o usuário. Nos próximos itens veremos através de um fluxograma de lógica como funciona cada arquivo.

6.3.1 Criação de Novos Arquivos

Um fluxograma da lógica para a obtenção do sistema "criar novo arquivo" é mostrado na figura 6.2.

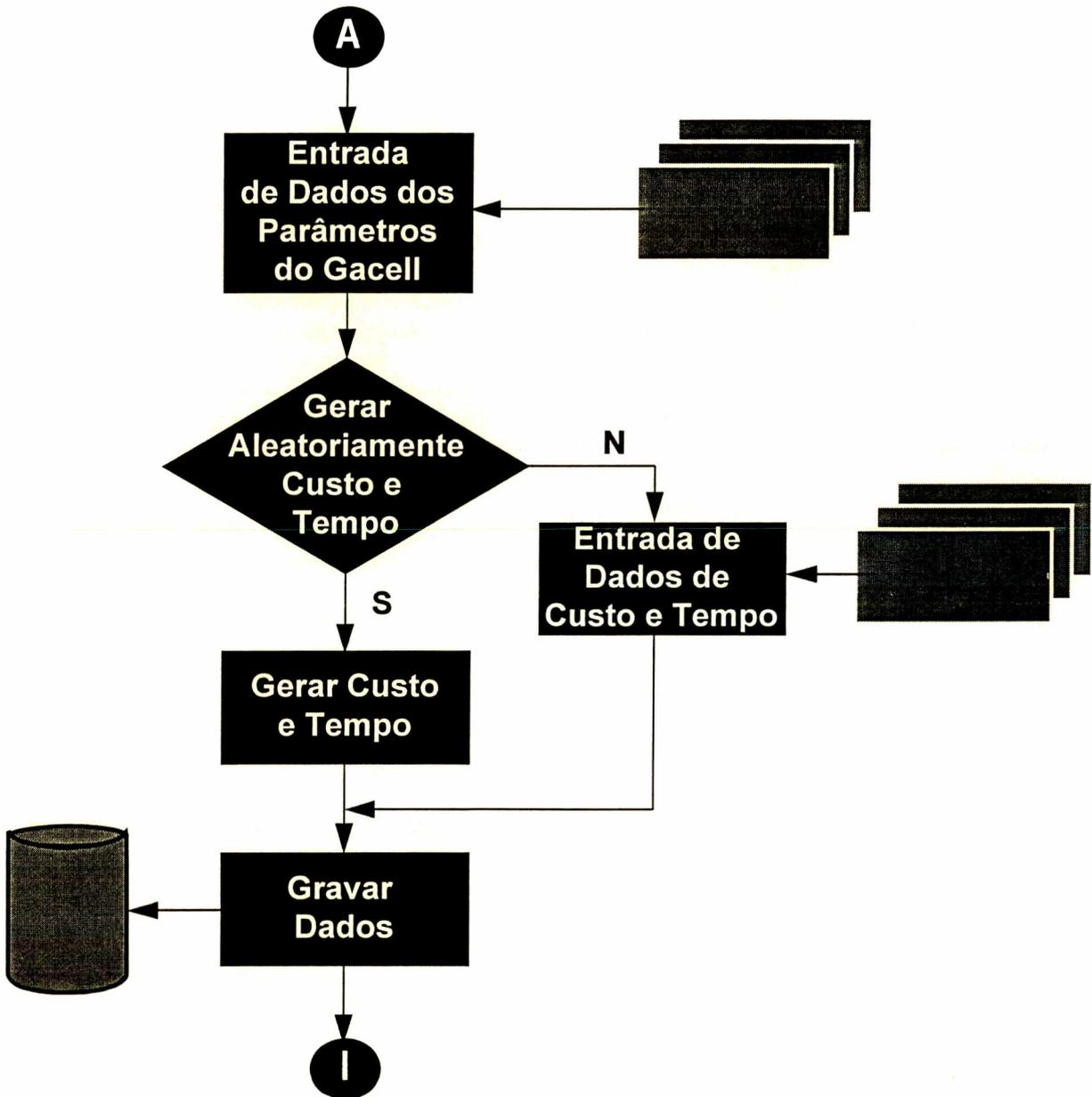


Figura 6.2 : Fluxograma de Lógica do Sistema "Criar Novo Arquivo".

A primeira pergunta feita "1 - Criar um novo arquivo" direciona o usuário ao ambiente dos arquivos. O programa pede ao usuário o nome do novo arquivo a criar (Anexo II). O usuário o fornece e o novo arquivo aparece na lista dos arquivos existentes. Aparece no toque da tecla **Enter** a segunda janela,

mostrada no Anexo III. Nesta janela aparecem duas opções, "1 - Digitar apenas os valores do Algoritmo Genético e 2 - Digitar todos os valores".

6.3.2 Alterar Arquivo Existente

As informações requeridas para a alteração do arquivo existente são mostradas no fluxograma de lógica do sistema na figura 6.3. Caso o operador escolha a opção "2" ele deverá obrigatoriamente entrar em um arquivo já existente, alterar os parâmetros (Anexo IV). Após isto retorna a janela principal, isto é, se desejar rodar o programa deve então utilizar a opção "3", mostrado no Anexo I

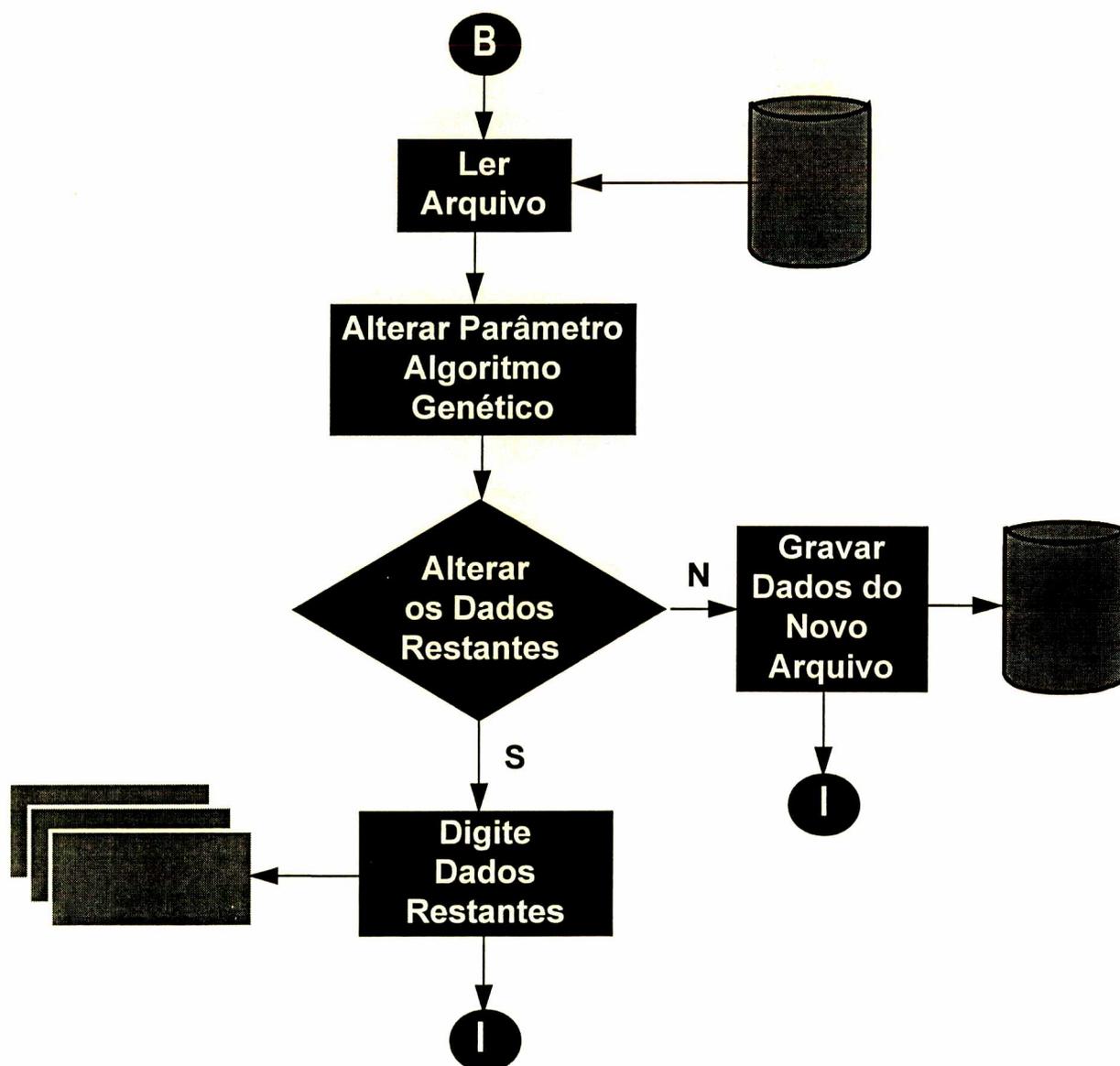


Figura 6.3 : Fluxograma de lógica do sistema "alterar arquivo".

6.3.3 Rodar Arquivo

Definidos os arquivos de trabalho, passa-se à etapa seguinte que é a formação das famílias utilizando o modelo apresentado no item 5.1. O fluxograma de lógica do sistema "rodar arquivo" na figura 6.4 mostra com detalhes a implementação do módulo da formação de família de peças.

A opção 3 "rodar arquivo" apresentada no Anexo II deverá ser usada quando o usuário já tiver parte dos arquivos de dados.

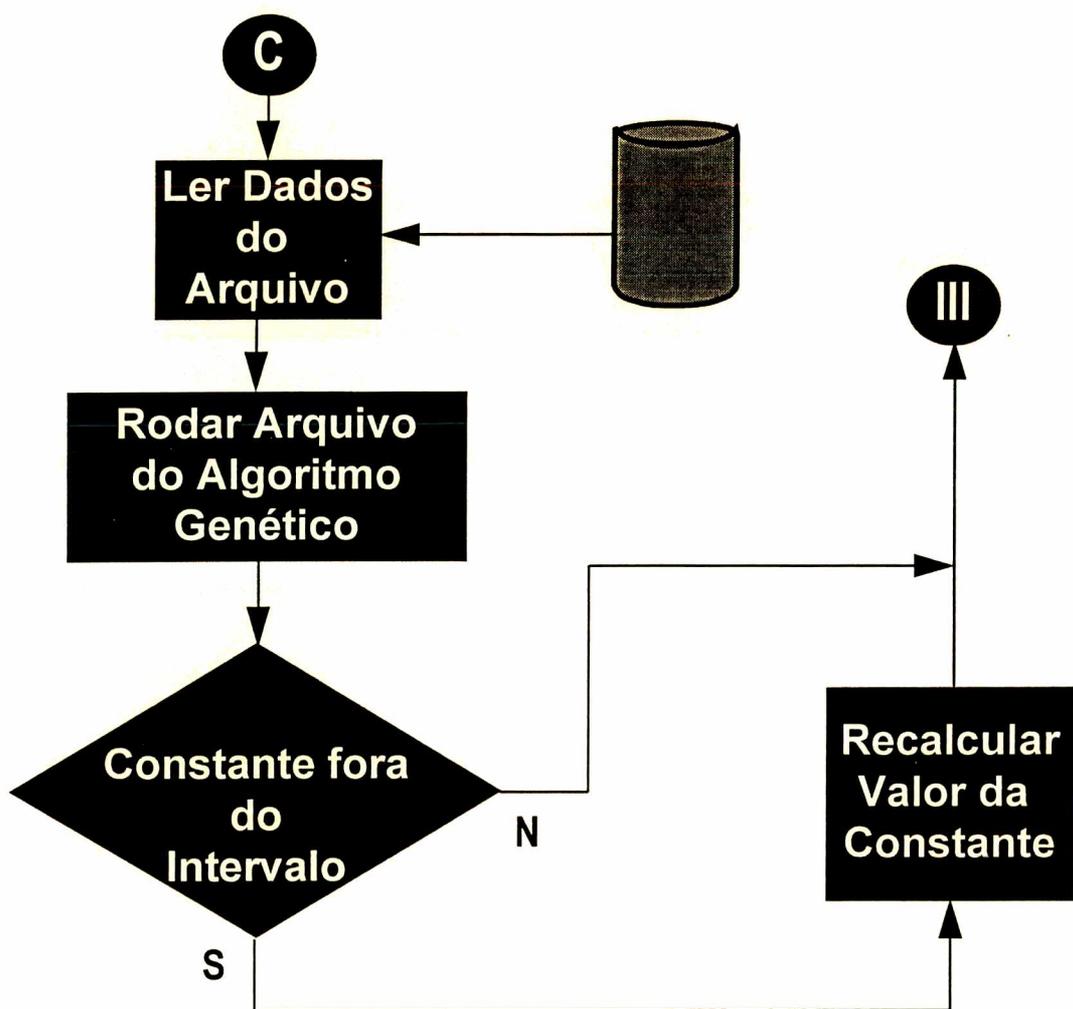


Figura 6.4 : Fluxograma de Lógica do Sistema "Rodar Arquivo".

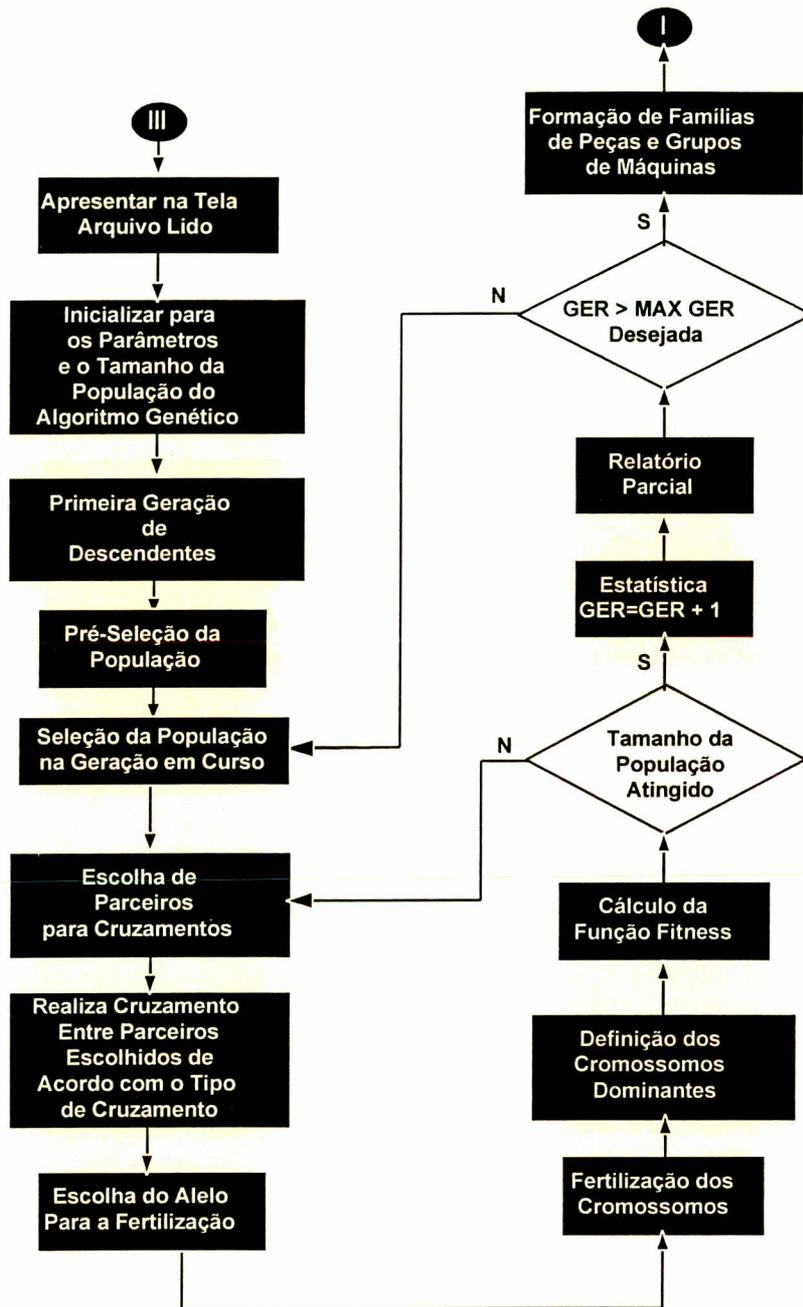


Figura 6.4 (continuação) : Fluxograma de Lógica do Sistema "Rodar Arquivo".

6.3.4 Apagar Arquivos

O fluxograma de lógica do sistema "Apagar Arquivo" é mostrado na figura 6.5.

Verificando o Anexo II, caso o operador queira apagar arquivos indesejáveis, deve utilizar a opção "4" e a opção "5" para sair do programa.

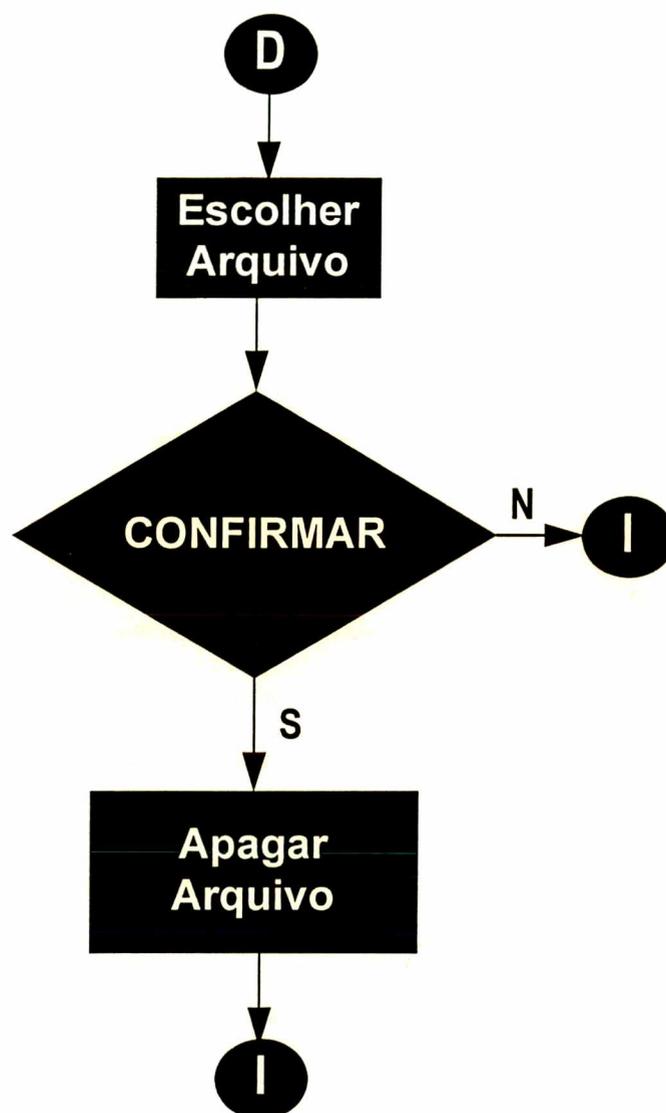


Figura 6.5 : Fluxograma de Lógica do Sistema "Apagar Arquivo".

6.4 Criar um Problema para a Simulação

Em princípio, seria o ideal possuir um problema para rodar o programa computacional, isto não foi possível, assim sendo, criou-se problemas hipotéticos e por intermédio deles calibrou-se o código computacional. Para ser criado o problema estabeleceu-se um número fixo de peças ($n_{peças}$), um número fixo de máquinas (n_{maq}), um número máximo de grupos (max_grupos), um número máximo de peças por grupo ($max_peça_grup$), e um número máximo de máquinas por grupo (max_maq_grup). A seguir, obedecendo a ordem numérica de 1 até o $n_{peças}$, é dado o número de planos

de fabricação para cada peça e a quantidade de peças a serem produzidas de cada tipo. Na seqüência, o número de máquinas disponíveis de cada tipo, o número de ferramentas disponíveis de cada tipo, o custo de capital de cada máquina. O tempo disponível para cada uma das ferramentas deve ser dado na ordem numérica de 1 até o número máximo de máquinas (n_{maq}).

Na figura 6.6 mostra-se uma listagem do computador dos dados de um problema. Podemos observar ainda nesta figura que o número de peças é 15 e que o número de máquinas é 10. Cada peça possui diversos planos de execução, ou seja, a peça "1" possui "2" planos, a peça "2" também "2" planos e a peça "3" possui "3" planos, etc...No caso das máquinas, por exemplo da máquina 9, há duas delas, com 5 ferramentas cada, e seu custo financeiro é de 250.00 unidades financeiras e seu tempo disponível é de 27000 unidades de tempo

GENETIC ALGORITHM FOR CELL FORMATION	
GACELL - Versão 1.0	
PARÂMETROS DO GACELL	
Nome do arquivo de saída (nome_arq)----->	yx1.out
Tamanho da população (tam_pop)----->	20
Número de peças (n_peças)----->	15
Número de máquinas (n_maq)----->	10
Número max. de grupos (max_grupos)----->	3
Número max. de peças/grup. (max_peças_grup)----->	5
Número max. de maq./grup. (max_maq_grup)----->	5
Geração máxima----->	2000
Probabilidade de cruzamento (prob_cruz)----->	0.90
Probabilidade de mutação (prob_mut)----->	0.0200
Dispersão máxima entre os indivíduos----->	0.0100
Limite inferior de utilização do grupo----->	0.30
Limite superior de utilização do grupo----->	0.95
Custo ótimo esperado----->	75000.00
Tolerância do custo esperado----->	0.020

Figura 6.6 : Dados para Rodar o Algoritmo Genético - Exemplo Hipotético

Genetic Algorithm for Cell Formation
GACELL - Versão 1.0

Parâmetros do GACELL

Número de planos por peça

2 2 3 1 2 1 1 2 1 1 2 2 2 1 1

Quantidade de peças a serem produzidas de cada tipo

10 12 15 5 9 8 18 7 4 12 3 14 20 6 11

Número de máquinas disponíveis de cada tipo

3 3 3 2 2 2 3 3 3 3

Número de ferramentas disponíveis por máquina

2 6 5 6 7 6 6 4 5 6

Custo de capital de cada máquina

100.00 150.00 200.00 250.00 300.00 100.00 150.00 200.00 250.00 300.00

Tempo máximo de disponibilidade de cada máquina

27000.00 27000.00 27000.00 27000.00 27000.00 27000.00 27000.00
27000.00 27000.00 27000.00

Máquinas ferramentas por peça, plano

(1,1) = (2, 2, 1.00, 4.50); (4, 1, 81.94, 7.24); (4, 2, 26.65, 14.75); (4, 3, 30.96, 6.59); (4, 4, 35.99, 10.81); (8, 1, 6.71, 11.60); (8, 2, 7.63, 17.45); (8, 3, 6.61, 6.69); (9, 1, 87.22, 9.89); (9, 2, 73.82, 9.25)

(1,2) = (1, 1, 66.58, 17.51); (4, 1, 66.49, 6.91); (4, 2, 16.29, 9.27); (4, 3, 44.81, 7.95); (8, 1, 76.61, 6.46); (8, 2, 46.29, 16.39); (9, 1, 83.19, 6.60); (9, 2, 73.64, 19.62); (9, 3, 47.30, 16.21); (9, 4, 76.76, 4.32);

(2,1) = (1, 2, 14.26, 6.30); (1, 3, 46.08, 4.35); (1, 4, 56.74, 4.15); (2, 1, 73.80, 14.41); (3, 1, 73.43, 15.33); (3, 2, 53.41, 14.30); (9, 1, 65.02, 13.49); (9, 5, 90.81, 14.30); (10, 1, 94.85, 7.90); (10, 2, 64.56, 6.73); (10, 3, 9.07, 16.36); (10, 4, 47.56, 16.11)

Na mesma listagem mostrada na figura 6.6 aparecem os dados associados à peça, plano, máquina, ferramenta, tempo e custo. Por exemplo, a informação $(1,1) = (2, 2, 1.00, 4.50); (4, 1, 81.94, 7.24)$... indica que a peça "1" executada sob plano "1" usa a máquina "2" na primeira operação, usando a ferramenta "2" com o tempo de "1" unidade de tempo e custo "4.50" unidades de custo, na segunda operação usa a máquina "4", a ferramenta "1" com tempo de "81.94" unidades de tempo e custo de "7.24" unidades de custo, etc...Estes dados, devido o seu volume, podem ser gerados aleatoriamente e esta opção pode ser utilizada no programa pelo usuário. Vimos por que um problema pode ser criado de uma forma totalmente arbitrária. Porém o programa na sua forma atual de desenvolvimento aceita perfeitamente a entrada de dados reais. Desta forma o usuário tem várias opções, conforme mostra a janela 1, ver Anexo I . Pode o operador criar novo arquivo, alterar arquivo, rodar arquivo já existente, apagar arquivos e sair do programa. A análise dos resultados, assunto do próximo parágrafo, deve ser feita via relatório de saída. Estes relatórios podem ser feitos em forma de tabelas ou gráficos. Nos gráficos pode fazer-se uma análise qualitativa do desempenho do algoritmo. As figuras 6.7 e 6.9 são uma amostra de um relatório gráfico. Na escala vertical aparece o custo obtido em cada geração, e na horizontal, o número de gerações. Vê-se claramente neste exemplo que rapidamente o algoritmo atinge a vizinhança do ponto ótimo (75.000 unidades financeiras), degenera algumas vezes, e retorna à vizinhança do ótimo esperado e continua próximo a ele. A figura 6.8 é a ampliação da figura 6.7 e a figura 6.10 é uma ampliação da figura 6.9, na qual se pode observar com mais clareza que após 200 gerações o ponto ótimo começa a ser alcançado, inclusive na geração "1154" é praticamente obtido, o programa degenera, retorna e a tendência geral na direção do ótimo é indicada pela linha tracejada. Na realidade, a figura 6.9 sugere que para um número muito grande de gerações o programa degenera. Embora, aparentemente, a primeira idéia é de que quanto maior o número de gerações, mais próximo chegaremos ao ponto ótimo. Veremos ao longo da discussão dos resultados que ela é falsa, pois quanto maior o número de gerações maior é a probabilidade de ocorrer degeneração, conforme indica as figuras 6.7, 6.8, 6.9 e 6.10, onde, por várias vezes, para um número de gerações maior que 1800, a função afastou-se do ponto ótimo.

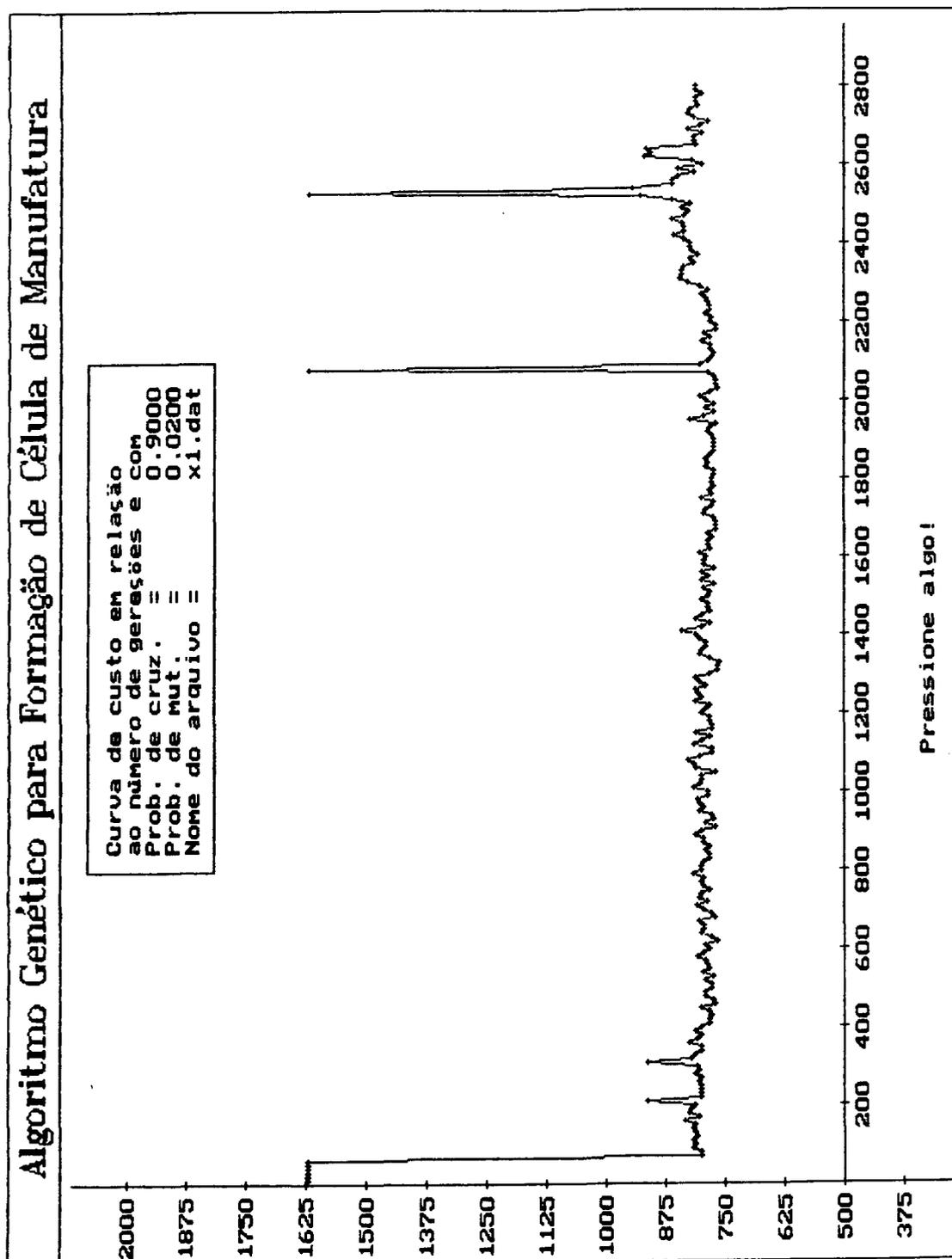


Figura 6.7 : Curva de custo em relação ao número de gerações

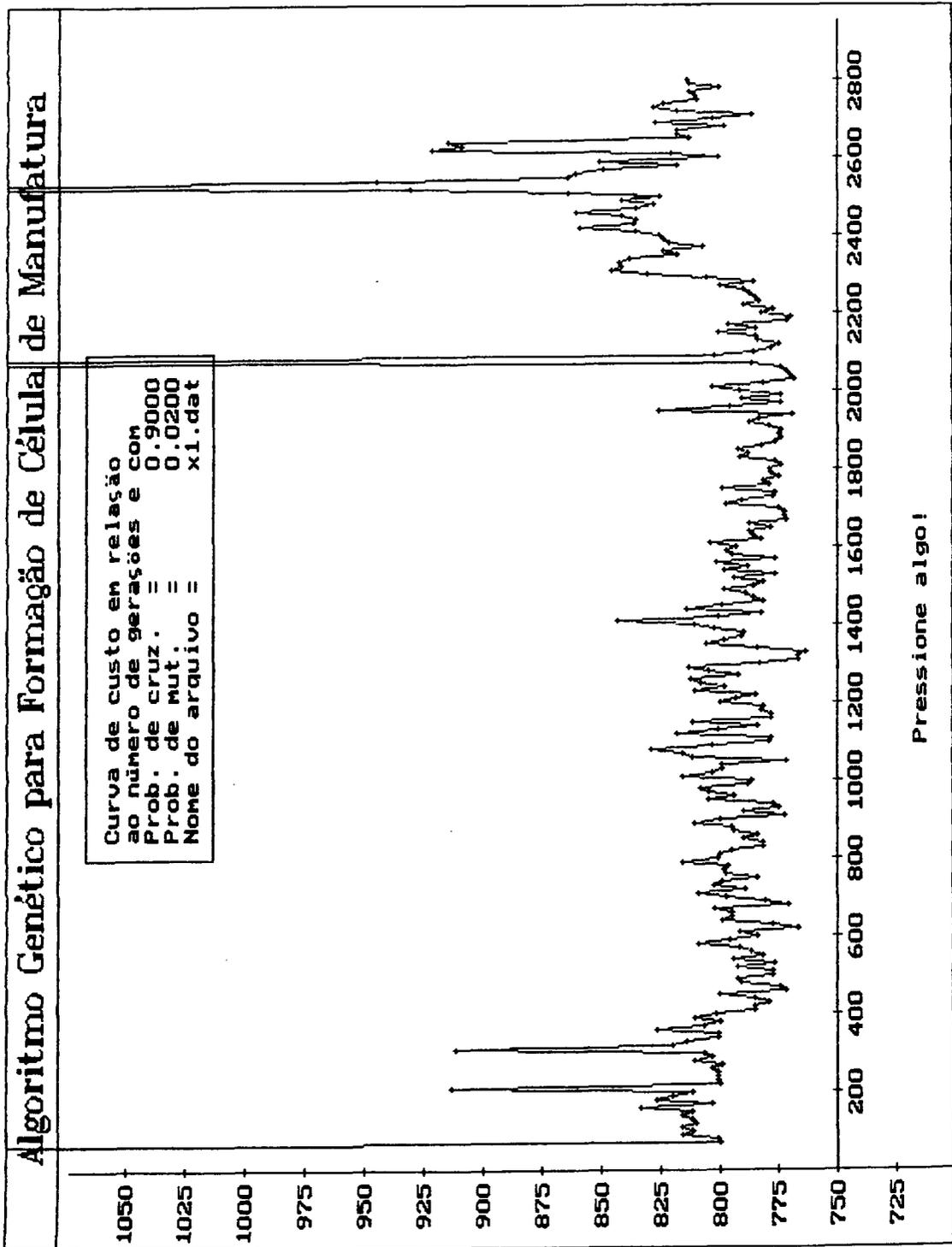


Figura 6.8 : Curva de custo em relação ao número de gerações

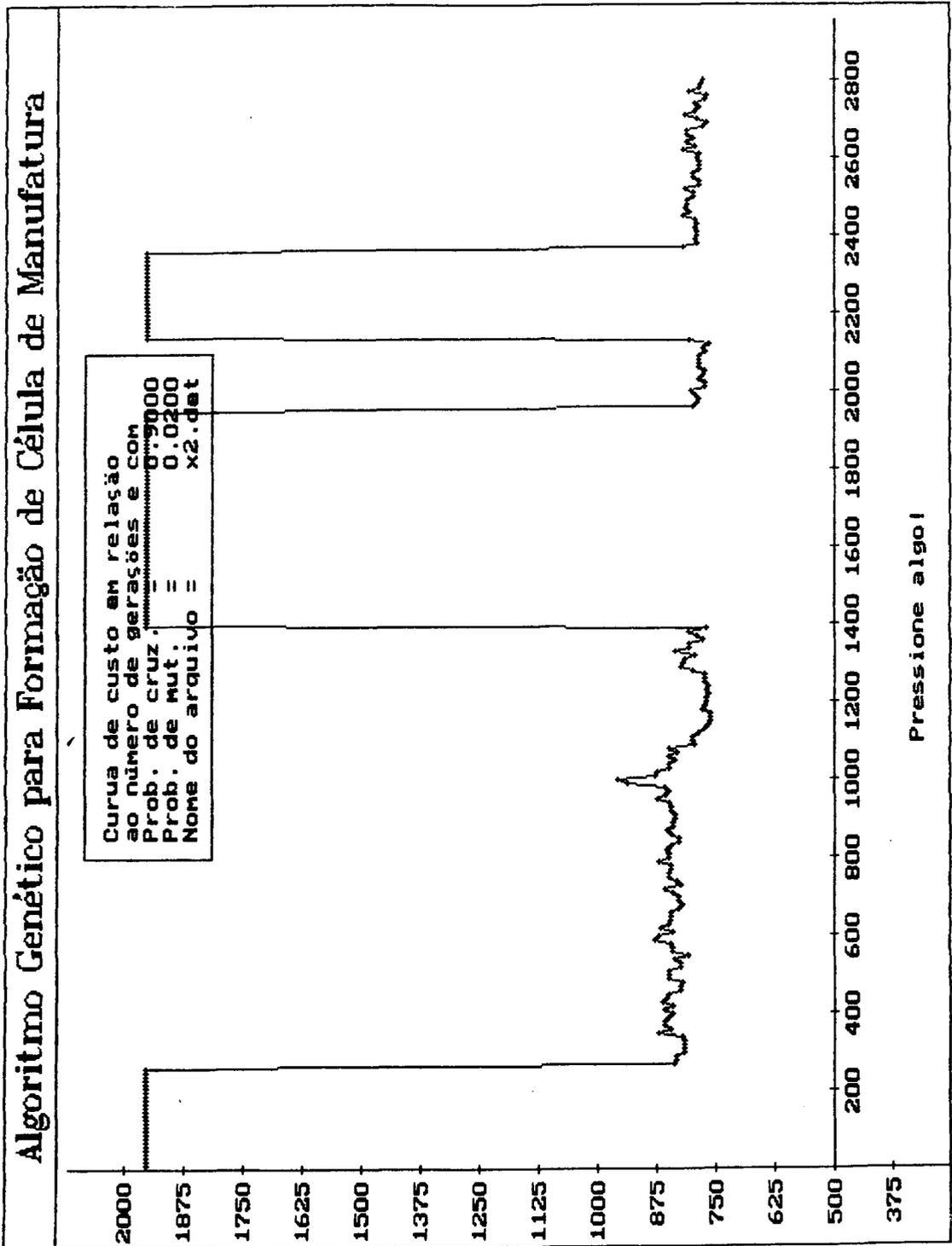


Figura 6.9 : Curva de custo em relação ao número de gerações.

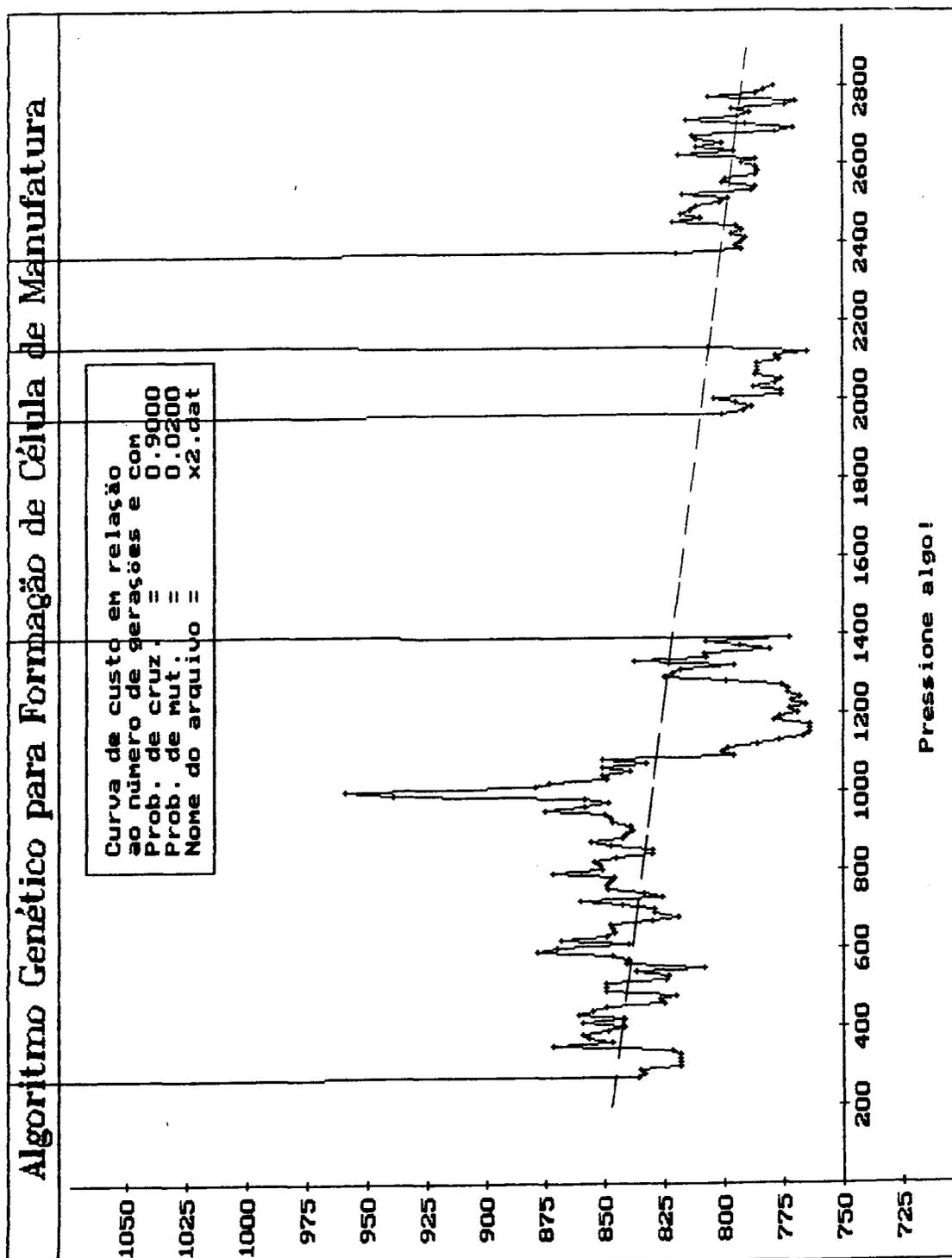


Figura 6.10 : Curva de custo em relação ao número de gerações.

6.5 Determinação dos Parâmetros do Algoritmo Genético

Os parâmetros fundamentais do Algoritmo Genético são: o número de elementos que compõem a população, isto é, seus cromossomos (formados segundo o código criado neste trabalho no capítulo 5), o número de gerações, ou seja, quantas vezes os cromossomos iniciais irão se reproduzir, a probabilidade de cruzamento e a probabilidade de mutação. A determinação dos valores mais adequados destes parâmetros é um processo exaustivo de procura e que será feito por tentativas e erros. Usando um micro computador 486 foram feitas aproximadamente 8000 soluções de um problema hipotético, cujos dados de peças, planos e máquinas foram obtidos do trabalho de Lashkari, R.S. e Gunasingh, K.R. (1990). Foi considerado pelo autor um Sistema Flexível de Manufatura com 15 peças, 10 máquinas e 3 grupos. Os planos de processo para cada peça e as correspondentes ferramentas utilizadas estão representados na tabela 6.1. Na tabela 6.2 são apresentadas as ferramentas a serem utilizadas em cada máquina e o número de tipos de máquinas disponíveis. A parte de tempos e custos foi criada de forma aleatória dentro de um intervalo, cujos valores destes intervalos foram tirados do trabalho de Jain, A.K., Kasilingam, R.G. e Bhole, S.D. (1990), usando o código computacional aqui criado. Como não existem regras formais para este processo de procura, esta tarefa é tediosa e longa. Cada solução leva em média de três a quatro minutos, fora a impressão de relatórios numéricos ou gráficos. Em tempo corrido, eliminando paradas, foram gastas 600 horas de computação. Todavia, a escolha adequada destes parâmetros é de fundamental importância para obter um método de cálculo mais apropriado, e com melhor uso de tempo de computação. Com esta idéia em mente, foi elaborado o planejamento desta experiência numérica, com o objetivo fundamental de estabelecer o melhor conjunto de parâmetros para rodar o Algoritmo Genético (*GACELL*) aqui proposto. Antes de ser feito o experimento numérico, o programa foi rodado exaustivamente e aleatoriamente, variando seus parâmetros, um número grande de vezes. O menor valor encontrado foi de "75159.9500" unidades financeiras, a célula correspondente a este resultado é mostrada na figura 6.11. Nesta figura, o valor ótimo é alcançado na geração "1886", cujas peças (1, 2, 6, 11, 13) formam o grupo 1, as peças (3, 10, 12, 14, 15) formam o grupo 2, e as peças (4, 5, 7, 8, 9) formam o grupo 3. As máquinas (1, 4, 9, 10) estão no mesmo grupo, as máquinas (6, 7, 8) formam um grupo e as máquinas (2, 3, 5) formam outro grupo.

Tabela 6.1 : Planos de Processo e Ferramentas Requeridas pelas Peças

PEÇA	PLANO DE PROCESSO	FERRAMENTAS REQUERIDAS
1	1	B02,D01,D02,D03,D04,H01,H03,M01,M02
	2	A01,D01,D02,D03,H01,H02,M01,M02,M03,M04
2	1	A02,A03,A04,B01,C01,C02,M01,M05,R01,R02,R03,R04
	2	A01,A02,A05,A06,C04,M02,M03,M04,R05,R06
3	1	A01,B01,B02,B03,B04,E02,E03,E04,F01,F02,F03,F04,G05,R02,R03,R04,R06
	2	B01,E02,E03,E04,E05,F04,F05,G01,G02,H03,R01,R03,R04
	3	B01,B03,E05,E06,F03,F04,F05,G01,R01,R02,R04
4	1	A02,A04,A06,C02,C03,C04,C05,F01,G01,M01,M03,M05,R02,R04,R05,R06
5	1	A01,D01,D02,H01,H02,M01,M02,M03
	2	D01,D03,D04,E07,H01,H02,H03,H04,M02,M03
6	1	A01,A03,A06,C02,C03,C04,G06,M04,M05,R01,R02,R04,R06
7	1	A01,B02,B03,B05,B06,E01,E03,E05,E06,E07,F05,G01,G02,R01,R02
8	1	B01,C01,E01,E03,E04,E07,F01,F05,G03,G04,G05,G06,R01,R03,R06
	2	B01,B03,D01,E01,E05,F01,F03,F04,G01,G03,R01,R02,R04,R06
9	1	C01,D02,D03,D04,H02,H03,H04,M04,M05
10	1	A02,A04,A06,C01,C02,C03,C05,E06,G05,M01,M06,R01,R02,R04,R05
11	1	A01,A02,C01,C03,C04,E01,M01,M03,M04,M05,R01,R02,R03
	2	A01,A02,A03,A05,C02,C03,H01,M01,M02,M03,M04,R01,R06
12	1	A01,B01,B02,B03,C01,E01,E02,F01,F02,F03,F05,G01,G02,H01,R01,R02,R03
	2	B01,B03,B04,C01,C02,E01,F01,F02,F03,H01,R01,R02,R03,R04
13	1	A01,D05,D06,F06,H06,H03,H04,M01,M03,M04,M05
	2	A05,D03,D04,D05,D06,H01,H02,H04,M04,M05
14	1	A01,A02,C01,C02,C04,H01,M01,M02,R01,R02,R03
15	1	B01,B02,E02,E04,E05,F01,F03,G02,R01,R02,R04,R06

Tabela 6.2 : Ferramentas Disponíveis nas Máquinas

MÁQUINAS	QUANTIDADE DISPONÍVEL	FERRAMENTAS DISPONÍVEIS
1	3	A01,A02,A03,A04,A05,A06
2	3	B01,B02,B03,B04,B05,B06
3	3	C01,C02,C03,C04,C05
4	2	D01,D02,D03,D04,D05,D06
5	2	E01,E02,E03,E04,E05,E06,E07
6	2	F01,F02,F03,F04,F05,F06
7	3	G01,G02,G03,G04,G05,G06
8	3	H01,H02,H03,H04
9	3	M01,M02,M03,M04,M05
10	3	R01,R02,R03,R04,R05,R06

RESULTADO ÓTIMO																
Geração Maior	=	1886														
Núm. da peça	=	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Grupo da peça	=	1	1	2	3	3	1	3	3	3	2	1	2	1	2	2
Plano da peça	=	2	1	3	1	1	1	1	1	1	1	2	2	2	1	1
Núm. da máq.	=	1	2	3	4	5	6	7	8	9	10					
Grupo da máq.	=	1	3	3	1	3	2	2	2	1	1					
		2	2	2	3	2	3	1	1	2	2					
		3	1	1				3	3	3	3					
Ajuste	=	109375.4700							Custo total ótimo = 75159.9500							

Ger.=2000		Cruz.=143937					Mut.=57488					Med.=28196.9030				
Desv.P.=43075.90																

Figura 6.11 : Resultado Ótimo - Exemplo Hipotético

O valor ótimo encontrado nesta solução do problema foi de 75159.9500 unidades financeiras, o número de cruzamentos é três vezes superior as das mutações. Este procedimento deve ser feito para que se possa estabelecer um intervalo de aceitabilidade da solução. Como foi encontrado o valor 75159.9500, após exaustiva procura, aproximadamente 100 soluções do problema, este valor servirá de base para se proceder à análise dos parâmetros do Algoritmo Genético. Para tal fim rodou-se o programa nas diversas composições mostradas nas tabelas 6.3, 6.4, 6.5. Cada composição foi rodada para uma probabilidade de mutação de 1% e de cruzamento de 90%. Escolheu-se esta composição de probabilidade, pois após exaustiva análise de resultados verificou-se que dentro destes valores a solução não degenera com tanta freqüência. Para rodar o total de casos das tabelas 6.3, 6.4 e 6.5 seriam necessários resolver 1360 ($8 \times 17 \times 10 = 1360$) vezes o problema. Com um tempo médio de 5 minutos por rodada, dariam aproximadamente 114 horas de computação. Os casos foram escolhidos de tal forma a traduzir a melhor composição do número de indivíduos da população versus número de gerações. As diferentes composições POP \times GER foram rodadas conforme os resultados iam sendo obtidos, e, analisados estes, serviam de orientação para as composições seguintes. Como para uma determinada composição POP \times GER, o produto POP \times GER dá o número de vezes que a função objetivo foi calculada e testada, pode dizer-se que em cada experimento, este produto (POP \times GER) nos fornece o total das tentativas executadas. Analisando este total em relação a

quantas vezes o valor da função ajuste cai dentro de certo intervalo pré-definido, pode-se então ter uma idéia da melhor ou pior composição obtida. Três intervalos foram escolhidos para análise: menor que 77000.00, menor que 78000.00 e menor do que 80000.00. As tabelas 6.3, 6.4 e 6.5 correspondem a estas três situações com $\text{prob_cruz} = 0.90$ e $\text{prob_mut} = 0.01$.

Tabela 6.3 : Número de Soluções Ótimas Usando Diferentes Gerações e Populações para Função Ajuste < 77000.00

GERAÇÕES																			
	10	20	30	40	50	60	70	80	90	100	200	300	400	500	1000	2000	4000	P	
10	0							1	0	0	0	0	0	1	3	3	8	♦	O
20	0	0		0				0	0	0	3	2	1	3	4	8	4	♣	P
30			0			0		0	1	0	3	3	4	4	9	5			U
40	0	0		0		0	0	0	0	0	2	7	2	3					L
50	0	0	1	0	0	0	0	1	1	3	4	6	5						A
60	0	0	0	0	1	0	1	2	3	2	3	5							Ç
70	0	1	2	0	0	2	1	0	1	10	5								Ã
80	0	0		0	2			2		3									O

♦ Degenera ♣ Tempo Excessivo (2 horas)

Tabela 6.4 : Número de Soluções Ótimas Usando Diferentes Gerações e Populações para Função Ajuste < 78000.00

GERAÇÕES																			
	10	20	30	40	50	60	70	80	90	100	200	300	400	500	1000	2000	4000	P	
10	0							1	0	0	1	2	0	3	6	5	9	♦	O
20	0	0		0		0		1	1	2	3	7	1	8	9	9	8	♣	P
30			0			1		2	4	1	3	7	8	8	10	7			U
40	0	0		0		0	1	0	3	3	5	9	6	8					L
50	0	0	1	0	1	0	0	2	7	3	7	7	8						A
60	1	0		0	1	3	1	6	5	3	6	8							Ç
70	0	2	2	2	2	3	3	2	3	10	7								Ã
80	0	0						6											O

♦ Degenera ♣ Tempo Excessivo (2 horas)

Tabela 6.5 : Número de Soluções Ótimas Usando Diferentes Gerações e Populações para Função Ajuste < 80000.00

GERAÇÕES																		
	10	20	30	40	50	60	70	80	90	100	200	300	400	500	1000	2000	4000	P
10	0							1	1	0	2	2	2	6	7	6	9 ♦	O
20	0	1		0		0		1	1	2	6	8	3	9	10	10	10 ♣	P
30			1			2		3	5	3	6	9	10	9	10	10		U
40	0	2		0		2	2	2	4	4	9	10	10	10				L
50	1	0	2	0	1	1	2	2	8	5	8	10	9					A
60	1	0		0	4	4	3	7	8	6	9	9						Ç
70	0	2	2	4	4	8	4	2	5	10	10							Ã
80	0	1		4	3			9		9								O

♦ Degenera ♣ Tempo Excessivo (2 horas)

Para a tabela 6.5 onde a Função Ajuste deve obedecer a restrição de ser menor que 80000.00, para ser considerada aceitável, as casas preenchidas com algarismos de 0 a 10 correspondem ao número de vezes, em 10 soluções, que a Função Ajuste atingiu, pelo menos uma vez dentro da população, naquela geração, um valor aceitável. Por exemplo tomemos uma população de 50 indivíduos e deixemos que ela se reproduza 200 vezes. Na realidade estamos praticando 10000 tentativas de alcançar o ponto ótimo, existe porém a possibilidade de repetição deste valor, tanto na população como nas gerações, para evitar esta situação o programa fornece somente o valor ótimo de cada geração. Desta forma, o valor que aparece no cruzamento da linha 50 com a coluna 200, ou seja, 8, significa que o valor da Função Ajuste naquele experimento foi 8 vezes inferior a 80000.00 nas 10 tentativas de encontrar o ponto ótimo (custo mínimo). Uma inspeção superficial das tabelas, no que concerne ao número de vezes que em cada experimento o valor ótimo foi encontrado, pode levar-nos a conclusões errôneas, por esta razão é importante que examinemos todas as combinações possíveis de resultados. Pergunta-se: Como examinar estas tabelas? Fixemos nossa atenção, por exemplo, na primeira linha da tabela 6.5. A população tem um número de elementos fixos, no caso 10. Para um número de gerações pequenas, menor que 100 por exemplo, raramente um valor ótimo é encontrado. A medida que o número de gerações cresce, aumenta o número de vezes que o valor ótimo é atingido. No caso da função Ajuste menor que 80000.00, este valor cresce até 9 vezes para 4000 gerações, acontece que a partir desse valor começa a ocorrer degeneração nas populações e a função objetivo é penalizada. Mesmo para os casos da função Ajuste menor que 78000.00 e 77000.00 o caso se repete. Fazendo aqui um paralelo com o processo de seleção natural, teríamos uma espécie animal

paralelo com o processo de seleção natural, teríamos uma espécie animal vivendo em uma ilha isolada e com recursos limitados, como a população não pode crescer mais que um certo limite, visto os recursos serem escassos, ela tende a degenerar, a não ser que, por qualquer razão, ela consiga entrar em contato com elemento da mesma espécie vivendo em outro local. Caso semelhante na natureza supõe-se que tenha ocorrido com os habitantes do Arquipélago da Ilha de Páscoa no litoral do Chile. Raciocinando-se desta forma, conclui-se que o número de 2000 gerações limita esta população de 10 indivíduos.

Examinemos agora uma coluna, por exemplo a coluna correspondente a 100 gerações, visto haver um crescimento sensível de resultados da função Ajuste que satisfaz a condição de ser menor que 80000.00 (tabela 6.5). Conforme se esperava, aumentando a população cresce o número de vezes que a função Ajuste é inferior a 80000.00. Porém o aumento da população, 60×100 , ou seja, 6000 tentativas, corresponde a um valor menor de tentativas do que o caso 10×4000 , isto é, 40000 tentativas. Embora não ocorra ainda degeneração, todavia o número de informações trocadas no computador é muito maior, devido ao fato do número de cruzamentos e mutações ser funções exponenciais do número de operações. Este acréscimo exponencial das operações lógicas aumenta sensivelmente o tempo de computação, tornando o cálculo proibitivo. Por exemplo, para a população de 20 indivíduos, um número de 4000 gerações provocaria para executar o problema 10 vezes, um tempo de 3 horas aproximadas num micro-computador 486-DX e 33 MHz. O exame da última linha das tabelas nos indica que para uma população de 80 indivíduos atinge-se um valor ótimo por volta de 80, 90 ou 100 gerações, correspondendo a 6400 tentativas.

Após exaustivo exame das tabelas 6.3, 6.4 e 6.5, conclui-se que as linhas transversais, marcadas pelas flexas, dividem as tabelas em três zonas. A primeira zona, canto inferior esquerdo, população crescente e número de gerações insuficientes, não sendo portanto uma região recomendada para trabalhar com o Algoritmo Genético. Descendo ao longo da diagonal divisória há uma contínua melhora no desempenho do par POP \times GER, porém a região central superior mostra-se também insuficiente até a coluna de 80 gerações. A partir deste valor a melhoria de desempenho é perfeitamente discernível. Já no canto superior das tabelas começa a ocorrer degenerações e o tempo de computação cresce exponencialmente para cada tentativa. Como no canto superior direito das tabelas ocorre com maior freqüência as melhores combinações do par POP \times GER, foram examinados com maior detalhe esses valores. A região contida entre as colunas de 400 a 2000 gerações e limitada inferiormente pela linha da população de 30 indivíduos, é a região onde devemos escolher, sem dúvida, a combinação ideal. Para tal fim ao somar-se os

sucessos nas diversas populações mais promissoras, 10, 20 e 30 indivíduos, vemos que para todas as faixas da Função Ajuste a combinação POP = 20 e GER = 1000 é a mais adequada. Talvez até, para economizar tempo de computação, poderíamos usar a combinação POP = 20, GER = 500. Escolhido o par "POP × GER" devemos então partir para verificar qual o melhor par prob_cruz × prob_mut, ou seja, qual a melhor probabilidade de cruzamento versus probabilidade de mutação que deve ser usada com o par, população 20 e geração 1000 Na tabela 6.6, construída exaustivamente, foram rodadas várias vezes o programa, isto é, dez vezes para cada par prob_cruz × prob_mut.

O aumento da probabilidade de mutação, num problema discreto como o que está sendo resolvido, pode provocar uma variabilidade muito grande nos elementos da população. Esta variabilidade pode ser útil, provocando, assim, o aparecimento prematuro de um valor próximo do ótimo, mas, por outro lado, tende mais rapidamente a desestabilizar o problema. Já fixando-se a probabilidade de mutação e diminuindo-se a probabilidade de cruzamento há uma maior estabilidade na população, conforme mostra a coluna 1 da tabela 6.6. Porém não é muito clara esta tendência. Os valores de 0.90 a 0.70 parecem indicar um intervalo ótimo no funcionamento do Algoritmo. Na linha de probabilidade de cruzamento 0.5 ocorre que aparentemente o método parece comportar-se melhor. Este resultado todavia é uma amostra, visto que das mil gerações somente 10 são mostradas. Como temos uma população de 20 indivíduos serão ao todo 20000 tentativas de resultados. Os gráficos deste tipo de caso apresentam em forma de uma seqüência de degenerações, conforme mostrado nas figuras 6.7, 6.8, 6.9 e 6.10. Isto quer dizer que na maior parte do tempo de vida da população, ela desaparece e reaparece, oscilando muito, mais na forma degenerada. De forma semelhante, uma taxa de mutação muito alta, isto é, maior do que 0.3, o número de degenerações e o número de sucessos (valores aceitáveis) diminui. Fica então limitada a uma combinação de 0.95 a 0.8 a probabilidade de cruzamento, e entre 0.01 a 0.02 a probabilidade de mutação. Dentro deste intervalo pode-se então escolher o par "POP × GER" que forneça com rapidez e segurança o valor otimizado do problema em exame.

6.6 Avaliação do desempenho

O algoritmo de formação de célula de manufatura foi testado em dois casos apresentados na literatura.

O primeiro exemplo é o modelo de programação inteira 0-1 proposto por Kusiak (1987). Trata do agrupamento de peças e máquinas em

Tabela 6.6 : Probabilidade de Cruzamento × Probabilidade de Mutação

POP = 20 GER = 1000													
CRUZA- MENTO	× 10 CUSTO	MUTAÇÕES											
		0.01			0.02			0.03			0.05		
0.95	<77	3			8			2			1		
	<76.5	3	0	10	8	1	10	1	6	10	1	10	10
	<76	0			4			0			1		
0.90	<77	9			7			4			0		
	<76.5	7	0	10	5	0	10	2	9	10	0	10	10
	<76	3			1			1			0		
0.85	<77	7			8			5			1		
	<76.5	4	0	10	5	2	10	4	7	10	0	10	10
	<76	2			2			1			0		
0.80	<77	6			10			7			1		
	<76.5	4	0	10	5	0	10	3	6	10	0	10	10
	<76	1			1			0			0		
0.75	<77	8			6			7			2		
	<76.5	7	0	10	4	0	10	3	4	10	1	10	10
	<76	4			1			0			1		
0.70	<77	4			6			5			1		
	<76.5	2	0	10	5	1	10	3	5	10	0	10	10
	<76	0			0			1			0		
0.50	<77	6			9			10			3		
	<76.5	2	0	10	7	2	10	5	1	10	2	9	10
	<76	2			6			0			1		
		SUC	DEG	BOM	SUC	DEG	BOM	SUC	DEG	BOM	SUC	DEG	BOM

presença de planos alternativos de processos com a definição de um índice de Similaridade S_{ij} entre a peça "i" e "j" em termos de planos e processos. Neste exemplo numérico mostrado na figura 6.12, 4 máquinas e 5 peças com seus respectivos planos de processos são consideradas. O resultado da aplicação do Algoritmo Genético com a função Fitness está resumido na tabela 6.7. Na figura 6.13 temos o resultado final das famílias de peças com os seus respectivos planos de processo e os grupos de máquinas. Confrontando os resultados verificamos que coincidem com o apresentado pelo autor, sendo que, para a Função Objetivo, encontramos valores melhores com planos de processos diferentes, conforme apresentado na tabela 6.7.

		Número de Peças																																																										
		1	2	3	4	5																																																						
							Número de Planos de Processo																																																					
		1	2	3	4	5	6	7	8	9	10	11																																																
$ a_{ij} $	=	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> <td style="width: 10%; text-align: center;">1</td> <td style="width: 10%; text-align: center;">1</td> <td style="width: 10%; text-align: center;">1</td> <td style="width: 10%;"></td> <td style="width: 10%; text-align: center;">1</td> <td style="width: 10%; text-align: center;">1</td> <td style="width: 10%;"></td> <td style="width: 10%; text-align: center;">1</td> <td style="width: 10%;"></td> </tr> <tr> <td></td> <td></td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td></td> <td></td> <td style="text-align: center;">1</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td style="text-align: center;">1</td> <td></td> <td></td> <td></td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td></td> <td style="text-align: center;">1</td> <td></td> </tr> </table>														1	1	1		1	1		1				1	1	1			1					1				1	1					1	1	1	1					1	1	1		1	
			1	1	1		1	1		1																																																		
		1	1	1			1																																																					
1				1	1					1	1																																																	
1	1					1	1	1		1																																																		

Figura 6.12 : Matriz inicial máquina-peça

1	3	2	4	5	Peça
4	4	5	5	5	Família
2	2	2	2	2	Plano
					Máquina
1	1	0	0	0	2
1	1	0	0	0	4
0	0	1	1	0	3
0	0	1	1	1	1

Figura 6.13 : Matriz Final

Tabela 6.7 : Resultado computacional da figura 6.12

Ord.	Efetivo	Planos	Fitness
1)	4 5 4 5 5	3 2 2 2 2	24.0000
2)	4 5 4 5 5	3 2 2 2 2	24.0000
3)	4 5 4 5 5	2 2 2 2 2	28.0000
4)	4 5 4 5 5	1 2 2 2 2	24.0000
5)	4 5 4 5 5	2 2 2 2 2	28.0000
6)	4 5 4 5 5	3 2 1 2 2	24.0000
7)	4 5 4 5 5	2 2 2 2 2	28.0000
8)	4 5 4 5 5	2 2 2 2 2	28.0000
9)	4 5 4 5 5	3 2 2 2 2	24.0000
10)	4 5 4 5 5	3 2 2 2 2	24.0000
11)	4 5 4 5 5	2 2 2 2 2	28.0000
12)	4 5 4 5 5	2 2 2 2 2	28.0000
13)	4 5 4 5 5	1 2 1 2 2	24.0000
14)	4 5 4 5 5	1 2 2 2 2	24.0000
15)	4 5 4 5 5	1 2 2 2 2	24.0000
16)	4 5 4 5 5	3 2 1 2 2	24.0000
17)	4 5 4 5 5	3 2 2 2 2	24.0000
18)	4 5 4 5 5	1 2 2 2 2	24.0000
19)	4 5 4 5 5	3 2 2 2 2	24.0000
20)	4 5 4 5 5	1 2 2 2 2	24.0000

Ger.= 72 Cruz.= 2303 Mut.= 284 Med.= 25.2000 Desv.P.= 1.8903

Prob. Cruzamento = 0.8

Prob. Mutação = 0.01

O segundo exemplo é um modelo com peças, máquinas e ferramentas proposto por LASHKARI, R. S. e GUNASINGH, K.R.(1990). Um exemplo hipotético apresentado pelos autores foi usado para ilustração. O exemplo numérico considera 15 peças, 10 máquinas e 3 grupos. Os planos de processo para cada peça e as correspondentes ferramentas requeridas, como também o número de ferramentas disponível em cada máquina foram fornecidos (tabela 6.1, 6.2). Um resultado computacional, aplicando o Algoritmo Genético, é mostrado na figura 6.14, utilizando como probabilidade de cruzamento 0.90 e probabilidade de mutação 0.02, e foram feitas 4000

gerações. A quantidade dos pontos ótimos aumenta com o aumento das gerações.

Ordem maior = 4															
Num. da peça =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Grupo da peça =	1	1	3	2	3	3	2	2	1	2	1	3	1	2	3
Plano da peça =	2	2	1	1	2	1	1	2	1	1	2	1	2	1	1
Num. da maq. =	1	2	3	4	5	6	7	8	9	10					
Grupo da maq. =	2	3	2	2	3	3	2	1	1	2					
	1	0	0	0	0	0	0	0	0	1	3				
Ajuste =	245690														
Ger.=	2806	Cruz.=	202172	Mut.=	80640	Med.=	67369	Desv.P.=	88503						

Figura 6.14 : Exemplo de uma solução computacional

O valor encontrado neste exemplo foi de 245690, o número de cruzamentos é três vezes superior ao de mutações. O valor ótimo deve ser encontrado, o qual servirá de base para se proceder a análise dos parâmetros do Algoritmo Genético.

6.7 Conclusão

O algoritmo genético mostrou-se efetivo nas soluções apresentadas, e é um processo ótimo baseado na maneira com que a vida vem sendo reproduzida por intermédio de mutações ou cruzamentos de seus genes dando origem a novos indivíduos.

A maioria dos métodos de otimização requer um tempo computacional enorme, o que mostra que este algoritmo é particularmente útil na solução de problemas complexos em larga escala, pois soluciona-os em menor tempo computacional.

CAPÍTULO 7

7. - Análise dos Resultados

7.1 Considerações Gerais

Conforme discutido no capítulo 6, o Algoritmo Genético adaptado ao processo de formação de células de manufatura mostrou ser uma ferramenta útil e valiosa.

No processo de agrupamento das peças e máquinas, definiu-se uma função objetivo (no caso do Algoritmo Genético, Função Ajuste) que tem a forma de vários picos intercelulares, conforme mostrado nas figuras 6.7, 6.8, 6.9 e 6.10 do capítulo 6. Otimizar uma função deste tipo pelos métodos tradicionais é praticamente impossível, visto ser difícil calcular suas derivadas. O uso de métodos heurísticos e similares baseados em procura exaustiva levaria a tempos impraticáveis nos computadores atuais. Isto quer dizer que o número de operações seria infinitamente grande. No caso de termos, "N" peças sob "P_i" planos, "M" máquinas usando "F_i" ferramentas sob "C_i" custos e "T_i" tempos, o número de operações seria maior que $N \times P_i \times F_i \times C_i \times T_i$. Quer dizer que devemos desenvolver um método mais adaptado à dimensionalidade do problema. Ao ser examinado neste trabalho o Algoritmo Genético como candidato a preencher esta necessidade, tivemos o cuidado de ponderar estes parâmetros.

A primeira conclusão importante é que, como o Algoritmo Genético trabalha com uma população de indivíduos, no caso um conjunto de valores da Função Ajuste, ele diminui o tempo de procura. No caso de uma população de vinte valores arbitrários da Função Ajuste, na realidade estamos trabalhando simultaneamente com o valor desta função em vinte pontos distintos e arbitrários no seu domínio. Estes pontos estão submetidos ao controle das restrições impostas que mantêm os valores da função dentro de uma faixa aceitável, caso ele não tenha sentido é penalizada. A penalização corresponde aos casos em que a população entra em degenerescência. Neste caso, na maioria dos exemplos estudados, a população não consegue retornar a região de soluções aceitáveis. Verificamos também que os parâmetros mais importantes são o tamanho da população, o número de gerações, a probabilidade de cruzamento e a probabilidade de mutação. Neste sentido, verificamos que o número de elementos muito grande na população acarreta um número enorme de operações, tornando o método muito longo. Os resultados

indicam que uma população de 20 a 30 indivíduos, trabalhando com até 1000 gerações, é a combinação ideal de parâmetros e os resultados foram sempre de ótima qualidade. No tocante às probabilidades, conclui-se que a melhor combinação encontra-se na faixa de 0.9 a 0.7 para a probabilidade de cruzamento e de 0.01 e 0.02 para a taxa de mutação.

Na construção do método visou obter-se uma ferramenta de fácil uso e manejo, mesmo por pessoas não muito familiarizadas com o uso do computador, o que foi atingido plenamente visto o programa ser auto explicativo.

A parte da simulação fornece em forma de relatório final o comportamento, custo e disposição dos grupos de peças e máquinas, bem como o custo final mínimo da solução. Pode selecionar-se várias soluções, visto que trabalhamos sempre com uma população e cada indivíduo é isoladamente uma solução do problema. O programa está sempre, para cada geração, computando o menor valor dos valores apresentados. A título de ilustração, selecionamos os onze melhores valores da Função Ajuste, retirando os casos em que repetiu-se o melhor valor e os que degeneram.

Conclui-se que esses dez menores valores da Função Ajuste nos fornecem, então, dez opções de arranjos das células. No caso, 75157.95 corresponde ao mínimo dos mínimos.

Estes resultados nos levam a concluir que os objetivos propostos para este trabalho foram atingidos. O método é bastante simples, converge e é estável, não sofrendo restrições de qualquer natureza. Ele transforma o processo de criação de grupos de peças e máquinas na manufatura a uma tarefa simples, direta e flexível. Cria-se na literatura uma sistemática para o planejamento de células de fabricação usando o Algoritmo Genético via computador.

Valor	POP × GER	Prob.Cruz.×Prob. Mut	Geração.
75157.95	20 × 2000	0.9 × 0.02	1886
75254.91	20 × 2000	0.8 × 0.02	462
75342.93	20 × 1000	0.75 × 0.01	170
75427.07	20 × 1000	0.9 × 0.01	171
75434.28	10 × 4000	0.9 × 0.01	762
75441.66	20 × 1000	0.5 × 0.03	814
75443.16	20 × 300	0.9 × 0.01	299
75453.91	20 × 1000	0.75 × 0.01	544
75466.50	20 × 1000	0.5 × 0.05	501
75476.76	30 × 1000	0.9 × 0.01	644
75517.94	20 × 2000	0.8 × 0.02	770

7.2 Conclusões

Neste trabalho de pesquisa buscou desenvolver-se uma nova sistemática de se obter uma forma de agrupar máquinas-peças em células de manufatura envolvendo custos e tempos. Para tal fim usou-se o Algoritmo Genético. Parte importante, como contribuição original para esta área de ponta, foi a adaptação deste Algoritmo à formação de células de manufatura.

Esta metodologia apresenta algumas características ainda não encontradas em outros trabalhos, resumidas como:

- a) construção de um modelo que permita planos de processo alternativo e o balanceamento ótimo da célula de manufatura;
- b) a solução deste modelo utilizando Algoritmo Genético;
- c) o agrupamento simultâneo de peças e máquinas automaticamente sem a utilização do procedimento de partição em dois sub-modelos;
- d) agrupamento dinâmico onde as famílias de peças são automaticamente formadas sem intervenção do usuário;
- e) a qualidade do agrupamento obtido através da determinação dos valores apropriados dos parâmetros do Algoritmo Genético.

7.3 Sugestões para a Continuação do Trabalho

O objetivo deste trabalho foi a criação e implementação do Algoritmo Genético no processo de formação de células de manufatura. Não houve a preocupação de torná-lo totalmente operacional e pronto para a aplicação no setor de produção. A linguagem adotada de início foi Turbo Pascal, que tem sérias limitações de tamanho de memória, o que restringe a utilização do código do Algoritmo Genético a problemas de pequeno porte. Por este motivo o protótipo foi reprogramado em linguagem C⁺⁺ da Borland.

Sugere-se a continuidade de estudos a serem feitos na área de células de manufatura, sendo importante ressaltar que a aplicação do Algoritmo Genético em agrupamento de peças-máquinas tem sido limitada. A pesquisa mostrou que a aplicação do Algoritmo Genético na formação de célula de

manufatura é viável. Os resultados obtidos nos modelos tirados da literatura foram bastante realísticos. Sendo assim, alguns tópicos importantes de pesquisa para futuros estudos na área de célula de manufatura são os seguintes:

- a) comparação do Algoritmo Genético com outras técnicas quantitativas e não quantitativas;
- b) combinação do Algoritmo Genético com outras técnicas procurando a hibridização (inclusive com alguns modelos heurísticos);
- c) desenvolvimento de uma estrutura de banco de dados de maneira a integrar a Tecnologia de Grupo com os dados de processo (MRPII), a ordem de entrada dos dados e os requerimentos de chão de fábrica;
- d) desenvolvimento de um algoritmo de maneira a aproveitar um banco de dados CAD para automaticamente gerar os planos de processo através do desenho das peças;
- e) construção de um protótipo para acomodar rápidas mudanças apresentadas pelo chão de fábrica;
- f) levar em consideração no modelo a transferência de uma peça de longo processamento de uma célula sobrecarregada para uma outra com capacidade disponível;
- g) modificação do modelo levando em consideração, por exemplo o custo de manutenção e de movimentação de material;
- h) integração ao modelo o problema de seqüenciamento
- i) e, por último, do ponto de vista prático a implementação do modelo a um caso real.

Referências Bibliográficas

- AL-QATTAN, Ibrahim. Designing flexible manufacturing cells using a branch and bound method. Int. J. Prod. Res., v. 28, n.2, p.325-336, 1990.
- ALI, Adel L., ALI, Dia L. and ALI, Kamal S.. The undeterminist manipulation of solid models for robot program synthesis, Computer Ind. Engng., v.19, n.1-4, p.465-4468, 1990.
- ANDEBERG, M. R. Cluster Analysis for applications. New York: Academic Press, 1973.
- ANDEBERG, M. R. Cluster Analysis for applications. New York: Academic Press, 1973.
- BEDWORTH, David D., HENDERSON, Mark R., WOLFE, Philip M. Computer-integrated design and manufacturing. McGraw-Hill, 1991.
- BEDWORTH, David D., HENDERSON, Mark R., WOLFE, Philip M. Computer-integrated design and manufacturing. McGraw-Hill, 1991.
- BOW, Sing - Tze Pattern Recognition-Applications to Large Data-Set Problems. New York: Marcel Dekker, 1984.
- BOW, Sing - Tze Pattern Recognition-Applications to Large Data-Set Problems. New York: Marcel Dekker, 1984.
- BROWNE, Jimmie et al. Production management systems; a CIM perspective. New York : Addison Wesley, 1988.
- BROWNE, Jimmie et al. Production management systems; a CIM perspective. New York : Addison Wesley, 1988.
- BURBIDGE, J.L., Production Flow Analysis on the Computer.In: Third Annual Conference of the Institution of Production Engineers. 1973 b.

- BURBIDGE, J.L., Production Flow Analysis on the Computer. In: Third Annual Conference of the Institution of Production Engineers. 1973 b.
- BURBIDGE, J.L.. The Introduction of Group Tecnhnology .New York, John Wiley, 1975.
- BURBIDGE, J.L.. The Introduction of Group Tecnhnology .New York, John Wiley, 1975.
- BURBIDGE, John L. The introduction of group technology. London: Butler & Tanner Ltd., 1975.
- BURBIDGE, John L. The introduction of group technology. London: Butler & Tanner Ltd., 1975.
- BURBIDGE, John L.. Change to group technology: process organization isobsolete . Int. J. Prod. Res., v.30, n.5, p.1209-1220, 1992.
- CARPENTER G, GROSSBERG, S., Search mechanisms for resonance theory (ART) architecture. In: Proceeding of the 3rd IEEE Annual International Conference in Neural Networks, p.1201-1205, 1989.
- CARRIE, A. S. Numerical taxonomy applied to group technology and plant layout. International Journal of Production Research, v.11, n.4, p.399-416, 1973.
- CHAN, H. M., MILNER, D. A. Direct clustering algorithm for group formation in cellular manufacture, Journal of Manufacturing Systems, v.1 n.1, p.64-76, 1982.
- CHANDRASEKHARAN, M. P., RAJAGOPALAN, R. An ideal seed non-hierarchical clustering algorithm for cellular manufacturing Int. J. Prod. Res., v.24, n.2, p.451-464, 1986.
- CHANDRASEKHARAN, M. P., RAJAGOPALAN, R. Groupability: an analysis of the properties of binary data matrices for group technology. Int. J. Prod. Res., v.27, n.6, p.1035-1052, 1987.
- CHANDRASEKHARAN, M. P., RAJAGOPALAN, R. MODROC : an extension to rank order clustering. Int. J. Prod. Res., v.24, n.5, 1986b.

- CHANDRASEKHARAN, M. P., RAJAGOPALAN, R. ZODIAC - An algorithm for concurrent formation of part-families and machine-cells. Int. J. Prod. Res., v.25, n.6, p.835-850, 1987.
- CHOOBINCH, F. A framework for design of cellular manufacturing systems. Int. J. Prod. Res., v.26, n.7, p.1161-1172, 1988.
- CHU, C. H. Cluster analysis in manufacturing cellular formation. Omega International Journal of Management Sciences, v.17, n.3, p.289-295, 1989.
- CHU, Chao-Hsien, TSAI, Mayshing. A Comparison of three array-based clustering techniques for manufacturing cell formation . Int. J. Prod. Res., v.28, n.8, p.1417-1433, 1990.
- DAVIS, Lawrence. Handbook of Genetics Algorithms. New York: Van Nostrand Reinhold, 1991.
- DE BEER, C., GINDHOVEN, T. H., DE WITTE, J. Production flow analysis. In: Annals of the Cirp, v.27. n.1, 1978.
- DE BEER, C., VAN GERWEN, R., DE WITTE, J. Analysis of Engineering production systems as a base for product-oriented reconstruction. In: Annals of the Cirp, v.25. n.1, 1976.
- DE WITTE, J. The use of similarity coefficient in production flow analysis. In: 5th Int. Conf. Prod. Res., p.36-39, 1979.
- DEVRIES, M. F., HARVEY, S. M., TIPNIS, V. A., VIJAY, A. Group technology - an overview and bibliography. MDC Manufacturing Report, Publication N.MDC76-101, Metcut Research Associates Inc., august 1976.
- ELMARAGHY, H. A., GU,P. Feature based expert parts assignment in cellular manufacturing. Journal of Manufacturing Systems, v.8, n.2, p.139-152, 1990.
- FU, K. S. Recent developments in pattern recognition. IEEE Transactions on Computers, v.C-29, n.10, p.845-854, oct 1980.

- GOLDBERG, D. E. Genetic algorithms in search, optimization, and machine learning. Addison Wesley, 1989.
- GOLDHAR, Joel D., JELINEK, Mariann. Plan for economies of scope. Harvard Business Review, p.141-148, november/december 1983.
- GROOVER, M. P. Automation, Production Systems and Computer Integrated Manufacturing. New Jersey: Prentice Hill, 1987.
- GU, P. H., ELMARAGHY H. A. Formation of manufacturing cells by cluster-seeking algorithms. Journal of Mechanical Working Technology, v.20, p. 403-413, 1989.
- GUNASINGH, K. Raja, LASHKARI, R. S. The cell formation problem in cellular manufacturing systems - a sequential modelling approach. Computers Ind Engng., v.16, n.4, p. 469-476, 1989.
- HAM, I., HITOMI, K., TOKIDA, T., Group Technology: applications to production management Boston: Kluwer Nijhoff Publishing Co., 1985.
- HAN, Chingping, HAM, Inyong . Multiobjective cluster analysis for part family formation. Journal of Manufacturing Systems, v.5, n.4, p.223-230, 1987.
- HYER, Nancy Lea. The potencial of group technology for U.S. manufacturing. Journal of Operations Management, v.4, n.3, may 1984.
- JAKUBOSWSKI, R., KASPRZAK, A. A syntactic description and recognition of rotary machine elements. IEEE Transactions on Computers, v.C-26, n.10, p.1039-1043, october 1977.
- KING, J. R., NAKORNCHAI, V. Machine-component group formation in group technology: review and extension. Int. J. Prod. Res., v.20, n.2, p.117-133, 1982.
- KING, John R. Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. Int. J. Prod. Res., v.18,n.2, p.213-232, 1980.
- KLIMASAUSKAS, Casimir C.. Teaching your computer to learn: applications of neural computing. Neural Ware Incorporated, 1988.

- KUMAR, K. R., KUSIAK, A., VANELLI, A. Grouping of parts and components in FMSs. European Journal of Operational Research, v.24, n.2, p.387-397, 1986.
- KUSIAK, Andrew, CHUNG, Yunkung. GT/ART : using neural networks to form machine cells. Manufacturing Review, v.4, n.4, december 1991
- KUSIAK, Andrew. An expert systems for group technology. Industrial Engineering, p.89-93, october 1987.
- KUSIAK, Andrew. Artificial intelligence series, part 6: Designing expert systems for scheduling of automated manufacturing. Industrial Engineering, v.19, n.7, p.42-46, july 1987.
- KUSIAK, Andrew. EXGT-S: A knowledge based system for group technology. Int. J. Prod. Res., v.26, n.5, p.887-904, 1988.
- KUSIAK, Andrew. Flexible manufacturing systems: a structural approach. Int. J. Prod. Res., v.23, n.6, p.1057-1073, 1985.
- KUSIAK, Andrew. The generalized group technology concept Int. J. Prod. Res., v.25, n.4, p.561-569, 1987.
- KUSIAK, Andrew. The part families problem in flexible manufacturing systems. In: Annals of Operations Research, n.3, p.279-300, 1985.
- KUSIAK, Andrew. The production equipment requirements problem. Int. J. Prod. Res., v.25, n.3, p.319-325, 1987.
- LASHKARI, R. S., GUNASINGH K. R. Machine-Part grouping in the presence of alternate process plans in flexible manufacturing systems. In: PROCEEDINGS OF MANUFACTURING INTERNATIONAL ' 90, 1990, Atlanta, Georgia, USA.
- LI, J., DING, Z., LEI, W. Fuzzy cluster analysis and fuzzy recognition methods for formation of part families. In: 14th NORTH AMERICAN MANUFACTURING RESEARCH CONFERENCE, 1986.
- LIPPMANN, Richard P.. An introduction to computing with neural nets. IEEE ASSP Magazine, v.3, n.4, p.4-22, april 1987.

- LOGAR, B., PEKLENIK, J. Computer-aided selection of reference parts for GT part families. In: 19TH CIRP Manufacturing Systems Seminar, June30/July1 1987.
- McAULEY, John. Machine grouping for efficient production. The Production Engineer, p. 53-57, february 1972.
- McCORMICK, William T., SCHWEITZER, R.J., WHITE, T.W. Problem decomposition and data reorganization by a clustering technique. Operations Research, v.20, n.5, p.993-1009, sept/oct. 1972.
- MITROFANOV, S. P. Scientific principles of group technology. Boston: National Lending Library for Science and Tecnology, 1966.
- PEKLENIK, Habil J.,GRUM, J. Computer-aided design of the part spectrum data base and its application to the design and production. In: Annals of the CIRP, v.31, n.1, p.313-317, 1982.
- PEKLENIK, Habil J.,GRUM, J. Investigation of the computer aided classification of parts. In: Annals of the CIRP, v.29, n.1, p.319-323, 1980.
- PEKLENIK, Habil J.,GRUM, J., LOGAR, B. An integrated approach to CAD/CAPP/CAM and group technology by pattern recognition. In: 16th CIRP INTERNATIONAL SEMINAR ON MANUFACTURING SYSTEMS, 1984,Tokyo, Japão.
- PEREIRA, Vera Lucia Duarte do Valle. Aplicação do algoritmo genético ao modelo de Kusiak. In: Anais do XIII ENEGEP, v.II, p.1000-1005, outubro 1993, Florianópolis, Santa Catarina.
- PEREIRA, Vera Lucia Duarte do Valle. Aplicação do algoritmo genético em planos de processos alternativos: modelo de Lashkari e Gunasingh. In: Anais do XIV ENEGEP, outubro 1994, João Pessoa, Paraíba.
- RAJAMANI, D., SINGH, N., ANEJA, Y.P. Integrated design of cellular manufacturing systems in the presence of alternative process plans. Int. J. Prod. Res. ,v.28, n-8, 1990.
- SEIFODDINI, Hamid, WOLFE, P. M. Application of the similarity coeficient method in group technology. IIE Transactions, september, 1986.

- SEIFODDINI, Hamid. Comparison between the similarity coefficient method and machine-component group analysis in group technology applications. Computers Ind. Engng., v.17, n.1-4, p.609-613, 1989.
- SRINIVASAN, R., LIU, C. R., FU, K. S. Extraction of manufacturing details from geometric models. Computer & Inds. Engng., v.9, n.2, p.125-133, 1985.
- VOHRA, Tajinder , CHEN, Der-San, CHANG, Jason C., CHEN, Hui-Chuan. A network approach to cell formation in cellular manufacturing. Int. J. Prod. Res., v.28, n.11, p.2075-2084, 1990.
- WEMMERLOV, Urban, HYER, Nancy L. Procedures for the part family/machine group identification problem in cellular manufacturing. Journal of Operations Management, v.6, p.125 - 147,1986.
- WU, H. L., RAGHUNATH, V., BARASH, M. M. Design of a cellular manufacturing system: a syntactic pattern recognition approach. Journal of Manufacturing Systems, v.5, n.2, p.81-88, 1986.
- ZHU, J. Y., ZHANG, Y. Z. An expert system of the part classification. In: Annals of the CIRP, v.36, n.1, p.351-353, 1987.

BIBLIOGRAFIAS

- ANDERSSON, H., JONSSON, H., AXSATER, S. A simulation study of hierarchical production-inventory control. Orspektrum, v.2, p.79-89, 1980.
- ANDO, K. et al. An approach to computer integrated production management. Int. J. Prod. Res., v.26, n.3, p.333-350, 1988.
- APPLETON, Daniel S. Systems integration series, part 2: Organizing and managing information resources in the data- driven enterprise. Industrial Engineering, p.62-97, june 1986.
- APPLETON, Daniel S. The state of CIM. Datamation, v.30,-n.21, p. 66-72, dec. 1984.
- ARBEX, Rogerio, SAVRA, Carlos E. Experiências praticas na expansão de sistemas. Máquinas e Metais, p. 74-82, fevereiro 1990.
- ASKIN, Ronald G., CRESSWELL, Steven H., GOLDBERG, Jeffrey B. VAKHARIA, Asoo J. A Hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing. Int. J. Prod. Res., v.29, n.6, p.1081-1100, 1991.
- AXSATER, Sven, JONSSON, Henrik. Aggregation and disaggregation in hierarchical production planning. European Journal of Operational Research, v.17, p. 338-350, 1984.
- BALLAKUR, A., STEUDEL, H. J. A within-cell utilization based heuristic for designing cellular manufacturing systems. Int. J. Prod. Res., v.25, n.5, p.639-665, 1987.
- BARD, Jonathan F. A heuristic for minimizing the number of tool switches on a flexible machine. IEE Transactions, v.20, n.4, p. 382-391, december, 1988.
- BEAL, K. Integrated material logistics and continuous flow manufacturing. Int. J. Res., v.26, n.3, p. 351-373, 1988.

- BERENJI, Hamid R., KHOSHNEVIS, Behrokh. Use of artificial intelligence in automated process planning. Computers in Mechanical Engineering, p. 47-55, september 1986.
- BERRA, P. B., BARASH, M. M. A computerised algorithm for the planning and optimization of a manufacturing process. Computer Aided Design, v.3, n.3, p. 24-28, spring 1971.
- BHASKARAN, Kumar. Process plan selection. Int. J. Prod. Res., v.28, n.8, p.1527-1539, 1990.
- BHAT, M. V., HAUPT, A. An efficient clustering algorithm. IEEE Transactions on Systems, Man, and Cybernetics, v.SMC-6, n.1, p.61-64, 1976.
- BIEGEL, John E., DAVERN, James J. Genetic algorithms and job shop scheduling. Computer Ind. Engng, [199-].
- BILLO, Richard E. et al. Integration of a group technology classification and coding system with an engineering database. Journal of Manufacturing Systems, v.6, n.1, p. 37-45.
- BITRAN, Gabriel R. et al. Hierarchical production planning: A single stage system. Operations Research, v.29, n.4, p.718-743, july/august 1981.
- BITRAN, Gabriel R. Hierarchical production planning. A two-stage system. Operations Research, v.30, n.2, march/april 1982.
- BITRAN, Gabriel R. et al. A hierarchical approach for the planning of a complex production system. In: RITMAN, 1979.
- BLACHE, Klaus M. et al. Process control and people at general motors delta engine plant. Material flow & Inventory Control. Industrial Engineering, p. 24- 30, march 1988.
- BOE, Warren J., CHENG, Chun Hung. A close neighbour algorithm for designing cellular manufacturing systems. Int. J. Prod. Res., v.29, n.10, p.2097-2116, 1991.

- BOSE, Partha Protim. Simulation models factory floors. American Machinist, p. 49-53, august 1988.
- BROWN, David C. Knowledge and control for a mechanical design expert system. Computer, p.92-100, july 1986.
- BROWN, Gerald E. et al. Tables for determining expected cost per unit under MIC- STD-LOSD single sampling schemes. AIIE Transactions, p.135, june 1974.
- BURBIDGE, John L. Operation scheduling with GT and PBC. Int. J. Prod. Res., v.26, n.3, p.429-442,1988.
- BURBIDGE, John L. The simplification of material flow systems. Int. J. Prod. Res., v.29, n.3, p.339-347, 1982.
- BURBIDGE, John L.. Change to group technology: process organization is obsolete . Int. J. Prod. Res., v.30, n.5, p.1209-1220, 1992.
- CAULLIRAUX, H.M., NAVEIRO, R. M. Projeto e simulação de células de produção. Máquinas e Metais, p.56-66, setembro 1989.
- CEDERQVIT, A. Cutting tool development for automation and FMS. In: Annals of CIRP, 1986.
- CHAKRABORTY, M. Implementation of mechanized manufacturing cells in an existing production as an economical alternative to FMS(a case study). In: Annals of CIRP, 1986.
- CHAN, Ding-Yu , BEDWORTH, David D. Design of a scheduling system for flexible manufacturing cells. Int. J. Prod. Res., v.28, n.11, p.2037-2049, 1990.
- CHAN, Ding-Yu, BEDWORTH, David D. Design of a scheduling system for flexible manufacturing cells. Int. J. Prod. Res., v.28, n.11, p.2037-2049, 1990.
- CHANG, Tien-Chien, WYSK, Richard A. An introduction to automated process planning systems. Prentice-Hill, Englewood Cliffs, New Jersey, 1985.

- CHEN, C. L., VENUPATI, V. S., ALJABER, N.. An application of genetic algorithm for flow shop problem. European Journal of Operational Research, v.80, n.2, p.397-341, january 1995.
- CHU, Chao-Hsien, HAYYA, Jack C. A fuzzy clustering approach to manufacturing cell formation. Int. J. Prod. Res., v.29, n.7, p.1475-1487, 1991.
- CHUAN, J. Chen et al. Hierarchical planning: a case study. Operational Research, 1981.
- DARROW, Willian P., GUPTA, Jatinder N. D. Integrating group technology and MRP systems through lot sizing and scheduling. Computers Ind. Engng., v.16, n.2, p.287-296, 1989.
- DE, Suranjan, WHINSTON, Andrew B. A framework for integrated problem solving in manufacturing. IIE Transactions, p. 284-297, september 1986.
- DEISENROTH, Michael P. A physical model for research in manufacturing systems control .In: Manufacturing Engineering Education Industry Conference, 1985, Michigan.
- DORNFELD , David A. Neural network sensor fusion for tool condition monitoring. In: Annals of the CIRP, v.39, n.1, p. 101-103, 1990.
- DUB, M. New developments in FMS, CAD/CAM and PM integration. In: Annals of CIRP, 1986.
- DUTTA, S. P. et al. The cell formation problem in flexible manufacturing systems. In: Annals of CIRP, 1986.
- DUTTA, S. P., LASHKARI, R. S., NADOLI, G., RAVI, T. A heuristic procedures for determining manufacturing families from design-based grouping for flexible manufacturing systems. Computer & Indus. Engng., v.10, n.3, p.193-201, 1986.
- EDOSOMWAN, Johnson Aimie. A program for managing productivity and quality. Industrial Engineering, v.19, n.1, p. 64-68, jan 1987.

- ELMARAGHY, H. A., GU, P. Knowledge- based system for assignment of parts to machine cells. The International Journal of Advanced Manufacturing Technology, v.3, n.1, p. 33-44, 1988.
- ELMARAGHY, H. A., GU,P. Formation of manufacturing cells by cluster seeking algorithms. Journal of Mechanical Working Technology, v.20, p.403-413, 1989.
- ELMARAGHY, H. A., GU,P.H. Expert parts assignment in cellular manufacturing using pattern recognition. Int. J.Mech. Tools Manufac., v.28, n.4, p.503-514, 1988.
- ELMARAGHY, H. A., GU,P.H. Expert system for inspection planning. In: Annals of the CIRP,v.36, p. 85-89, 1987.
- ENDOW, J. et al. FMS for plastic working in Japan. In: Annals of CIRP, 1986.
- EVANS, M. C. An FMS for high variety-short cycle time components. In: Annals of CIRP, 1986.
- EW, A. Normann, ROSA, J. M. D. O processo de furação e otimizado pelo CAPP. Máquinas e Metais, p. 28-41, julho 1989.
- FEITNER, Charles E., WEINER, Steven A. CIMS series, part 19: Models, myths and mysteries in manufacturing. Industrial Engineering, p. 66-76, july 1985.
- FINDLAY, R. C. Flexible manufacturing system in operation. In: Annals of CIRP, 1986.
- FISHER, M. L. The lagrangean relaxation method for solving integer programming problem. Management Science, v.27, n.1, p.1-18, 1981.
- FORSYTHE, Diana E., BUCHANAN, Bruce G. Knowledge acquisition for expert systems: some pitfalls and suggestions. IEEE Transactions on Systems, Man, and Cybernetics, v.19, n.3, p. 435-442, may/june 1989.
- GALLAGHER, C. C., KNIGHT, W. A. Group technology. London: Butterworth & Co. (Publishers) Ltd., 1973.

- GEIDERS, Ludo F., Steelandt, Frank V. Van. Design and implementation of a production planning system in a rolling mill: a case study. AIIE Transactions, v.12, n.1, p. 54-58., [199-].
- GERWIN, Donald. Do's and don'ts of computerized manufacturing. Harvard Business Review, p.107-116, march/april 1982.
- GETTINGS, Marty. A computer- aided approach to manufacturing quality. Industrial Engineering, p.18-21, march 1990.
- GHOSH, Bimank K., WYSK, Richard A. Post- optimality of computer networks in flexible manufacturing systems. Computers Ind. Engng., v.16, n.1, p.45-53, 1989.
- GILL, Philipe, MURRAY, Walter and WRIGHT, Margaret H. Practical Optimization. London: Academic Pree, 1981.
- GOLDRATT, Eliyahu. Computerized shop floor scheduling. Int. J. Prod. Res.,v.26, n.3, p.443-455, 1988.
- GONCALVES FILHO, V., CHRISTIANO, Antonio Carlos. Projeto e implantação de células de manufatura. Máquinas e Metais, n.293, p.12- 18, maio 1990.
- GREENE, Timothy J., CLEARLY, Collen M. Células de manufatura racionalizam a movimentação de materiais. Movimentação e Armazenagem, p. 10-17, mar/abr 1989.
- GRILLO, Richard C. de Meiroz. Implantando um software de tecnologia de grupo. Máquinas e Metais, p.46-54, junho 1988.
- GRUM, J. et al. Design of the database for CAD based on group technology. Robotics & Computer-Integrated Manufacturing, v.4, n.1/2, p.49-62, 1988.
- GUERRERO, Hector H. Group technology: I. The essential concepts. Production and Inventory Management, p. 62-70, first quarter, 1987.
- GUERRERO, Hector H. Group technology: II. The implementation process. Production and Inventory Management, p. 1-9, second quarter 1987.

- GUNASCKARAN, I, VIRTANEN, I, YLI-OLLI, P. An investigation into the application of group technology in advanced manufacturing systems. Int. Journal of Computer Integrated Manufacturing., v.7, n.4, p. 215-228, 1994.
- GUPTA, M. C. et al. Expert scheduling system for a prototype flexible manufacturing cell: A framework. Comput. Opns. Res., v.16, n.4, p. 363-378, 1989.
- GUPTA, Tarun, SEIFODINI, Hamid. Clustering algorithms for the design of a cellular manufacturing systems - an analysis for their performance. Computer Ind. Engng., v.19, n.1-4, p.432-436, 1990.
- GUPTA, Tarun, SEIFODINI, Hamid. Production data based similarity coefficient for machine-component grouping decisions in the design of a cellular manufacturing systems. Int. J. Prod. Res., v.28, n.7, p.1247-1268, 1990.
- HAIDER, S., BANKS, Jerry. Simulation series, part 3, Simulation software products for analyzing manufacturing systems. Industrial Engineering, p.98-103, july 1986.
- HAM, Inyong. Aplicações da tecnologia de grupo para maior produtividade na fabricação. São Paulo: Escola de Engenharia de São Carlos, 1986.
- HAN, Min-Hong, MCGINNIS, Leon F. Throughput rate maximization in flexible manufacturing cells. IIE Transactions, v.20, n.4, p.409-417, december 1988.
- HANDBOOK OF INDUSTRIAL ENGINEERING. John Wiley & Sons, 1982.
- HARHALAKIS, G., NAGI, R., PROTH, J. M. An efficient heuristic in manufacturing cell formation for group technology applications. Int. J. Prod. Res., v.28, n.1, p.185-198, 1990.
- HENDERSON, M. R., MUSTI, S. Automated group technology part coding from a three-dimensional CAD database. Journal of Engineering for Industry, v.110, p.278-287, august 1988.

- HERAGU, Sunderesh S., KUSIAK, Andrew. Analysis of expert systems in manufacturing design. IEEE Transactions on Systems, Man, and Cybernetics, v.17, n.6, p. 899-912, november/december 1987.
- HERAGU, Sunderesh S., KUSIAK, Andrew. Machine layout problem in flexible manufacturing systems. Operations Research, v. 36, n. 2, p.258-268, march/april 1988.
- HERROELEN, W., GILS, Van A. On the use of flow dominance in complexity measures for facility lay out problems. Int. J. Prod. Res.,v.23, n.1,p.97-108,1985.
- HERZOG, Raymond E. The link between customer and engineer. Machine Design, v.46, n.17, p.113-116, july 1974.
- HINSON, Ray. Machine load unload automation is robot challenge. Industrial Engineering, p. 103-106, march 1984.
- HINSON, Ray. Study analyzes utilizing robots in palletizing. Industrial Engineering, p.21-24, february 1984.
- HJELM, S. Methods for simplification of flexible manufacturing production system complex. In: Annals of CIRP, 1986.
- HOLMGREN, Kathleen M. Workstation producer's new plant provides manufacturing flexibility to introduce or switch products. Industrial Engineering, p.32-35, march 1988.
- HU, Evan C., SPRAGUE, James C. Designing an effective inventory classification and coding system. Production and Inventory Management Journal, p. 53-56, fourth quarter 1987.
- HYER, Nancy Lea, SUBRIN, Rachel. Capabilities of group technology. Michigan: The Computer and Automated Systems Association.
- HYER, Nancy Lea, WEMMERLOV, Urban. Group technology in the U.S. manufacturing industry: a survey of current practices. Int. J. Prod. Res., v.27, n.8, p.1287-1304, 1989.

- HYER, Nancy Lea. Capabilities of group technology. The Computer and Automated Systems Association of SMC. [S.L.], 1987.
- IWATA, Kazuaki. Knowledge-based flexible part classification system for CAD/CAM, In: Annals of the CIRP, v.36, n.1, p.317-320, 1987.
- JAIKUMAR, Rannchandran. An operational optimization procedure for production scheduling. Comput & Ops Res., v.1, p.191-200, 1974.
- JAIN, A. K., KASILINGAM, R. G., E BHOLE, S. D.. Cell formation in flexible manufacturing systems under resource constraints. Computers Ind. Engng., v.19, n.1-4, p. 437-441, 1990.
- KARGER, Delmar W., COX, John L. marketing as design partner. Machine Design, v.53, n.9, p. 51-54, apr. 23, 1981.
- KHATOR, S. K., IRANI, S. A. Cell formation in group technology: A new approach. Computers Ind. Engng, v.12, n.2, p.131-142, 1987.
- KING, John R. Machine-component group formation in group technology. The Int. J. Mgm. Science, v.8, n.2, p.193-199, september 1979.
- KINNEY JR.,Hugh D., MCGINNIS, Leon F. Facilities & Material handling systems series: part 6. Manufacturing cells solve material handling problems. Industrial Engineering, p. 54-60, august 1987.
- KISTNER, K. P, SWITALSKI, M. Hierarchical production planning necessity, problems, and methods. Zor-Methods and Models of Operations Research, v.33, p. 199-212, 1989.
- KLEMMING, B. Implementing FMS- planning for CIM or creating Islands of automation. In: Annals of CIRP, 1986.
- KNIGHT, Donald O., WALL, Michael L. Using group technology for improving communication and coordination among teams of workers in manufacturing cells. Industrial Engineering, p.28-34, january 1989.
- KNIGHT, Kevin. Connectionist ideas and algorithms. Communications of the ACM, v.33, n.11, p.59-74, november 1990.

- KUMAR , C. S., CHANDRASEKHARAN, M. P. Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. Int. J. Prod. Res., v.28, n.2, p.233-243, 1990.
- KUMARA, S.R.T. et al. Expert systems in industrial engineering. Int. J. Prod. Res.,v.24, n.5, p. 1107-1125, 1986.
- KUMARA, Soundar et al. Artificial intelligence series, part 1: an introduction to artificial intelligence. Industrial Engineering, v.18, n.12, p. 9-20, dec 1986.
- KUSIAK, Andrew, CHOW, Wing S. Efficient solving of the group technology problem. Journal of Manufacturing Systems, v.6, n.2, p.117-124, 1987.
- KUSIAK, Andrew, HERAGU, Suderesh S. Group Technology. Computers in Industry, v.9, p. 83-91, 1987.
- KUSIAK, Andrew, HERAGU, Sunderesh S. Knowledge-based systems guides machine layout in flexible manufacturing systems. Industrial Engineering, p.48-53, november 1988.
- KUSIAK, Andrew, HERAGU, Sunderesh S. The facility layout problem. European Journal of Operational Research, v.29, p.229-251, 1987.
- LAW, Averill M. Simulation series, part 1: Introduction to simulation: A powerful tool for analyzing complex manufacturing systems. Industrial Engineering, p.46-63, may 1986.
- LI, Rong-kwel, BEDWORT, David D. A framework for the integration of computer-aided design and computer-aided process planning. Computer Ind. Engng, v.14, n.4, p.395-413, 1988.
- LIBERATORE, Matthew J., MILLER, Tan. A hierarchical production planning system. Interfaces, v.15, p.1-11, july/august 1985.
- LINDGKEN, J. Different techniques for automatic identification in FMS installations. In: Annals of CIRP, 1986.

- LIV, Richard C., RAMESH, Srinivasan. Generative process planning using syntactic pattern recognition. Computers in Mechanical Engineering, p. 63-66, march 1984.
- LOGENDRAN, Rasaratnam, WEST, Thomas M. A machine-part based grouping algorithm in cellular manufacturing. Computer Ind. Engng., v.19, n.1-4, p.57-61, 1990.
- LOGENDRAN, Rasaratnam. A model for duplicating bottleneck machines in the presence of budgetary limitations in cellular manufacturing. Int. J. Prod. Res., v.30, n.3, p.683-694, 1992.
- LOONEY, M. W. FMS in a high variety small batch environment. In: Annals of CIRP, 1986.
- MACHADO, Virgilio A. Cruz, TEIXEIRA, J. J. Pamies. Tecnologia de grupo. Tecnometal, p.7-12, [199-].
- MARION, David et al. Systems integration series, part 3: Developing a group technologie, Industrial Engineering, v.18, n.7, p.90-97, july 1986.
- MELKANOFF, Michal A. CIMS Series, part 11. The CIMS database: Goals, problems, case studies and proposed approaches outlined. Industrial Engineering, v.16, n.11, p. 78-93, november 1984.
- MEREDITH, Jack. Automation strategy must give careful attention to the firm's infrastructure. Industrial Engineering, p. 68-73, may 1986.
- MITTAL, Sanjay et al. Pride: an expert system for the design of paper handling systems. Computer, p. 102-114, 1986.
- MOSIER, Charles T. An experiment investigation the application of clustering procedures and similarity coefficients to the GT machine cell formation problem. Int. J. Prod. Res., v.27, n.10, p.1811-1835, 1989.
- NAV, Damas, CHANG, Tien-Chien. Prospects for process selection using artificial intelligence. Computers in Industry, v.4, p. 253-263, 1983.
- NAWAZ, Muhammad et al. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega, v.11, n.1, p.91-95, [199-].

- NAYLOR, Arch W., KOLZ, Richard A. Design of integrated manufacturing system control software. IEEE Transactions on Systems, Man, and Cybernetics, SMS, v.17, n.6, p. 891-897, dec. 1987.
- NELSON, Craig A. A scoring model for flexible manufacturing systems project selection. European Journal of Operational Research, v.24, p.346-359, 1986.
- NOVAES, Antonio Galvão. Métodos de otimização: aplicações aostransportes. São Paulo: Edgard Blücher, 1978.
- PARKS, Michael W. Artificial intelligence series, part 1: Expert systems: Fill in the missing link in paperless aircraft assembly. Industrial Engineering, v.19, n.1, p.37-45, january 1987.
- PEKLENIK, Habil J. Development of the part spectrum database for computer integrated manufacturing systems (CIMS). In: Annals of the CIRP, v.39, n.1, p.471-473, 1990.
- PEKLENIK, Habil J. Report of the 19th CIRP International Seminar on Manufacturing Systems, p.255-265, 1988.
- PIERSON, Robert A. Adapting horizontal material handling systems to flexible manufacturing setups. Industrial Engineering, p.62- 71, march 1984.
- PLASSL, Keithr. Production in the factory of the future. Int. J. Prod. Res., v.26, n.3, p.501-506, 1988.
- QUINN, John J. Brief Case: strategic management of R&D. Long Range Planning, v.23, n.1, p. 147-150, 1990.
- RADOS, Gregório Jean Varvakis. Tecnologia de grupo, uma filosofia de trabalho. Máquinas e Metais, p. 35-43, junho 1986.
- RAJAGOPALAN, R., BATRA, J. L. Design of cellular productions system: a graph-theoretic approach. Int. J. Prod. Res., v.13, n.6, p.567-579, 1975.
- ROZENFEID, Henrique. Um sistema CAPP modular para indústrias pequenas e médias. Máquinas e Metais, p.26-38, março 1990.

- RUBIN, Michael G. A microcomputer-based materials management system for small manufacturing firms. Industrial Engineering, p. 54-59, june 1986.
- SABUNCUOGLU, Ihsan, HOMMERT, Zhein. Don. Computer usage in manufacturing systems. Computers Ind. Engng. v.15, n.1-4, p.1-7, 1988.
- SCUNK, Dan L., FILLEY, Richard D. Systems integration series, part 1: Systems integration's challenges demand a new breed of industrial engineer. Industrial Engineering, p.65-67, may 1986.
- SEIDMANN, Abrahan, SCHWEITZER, Paul J. Part selection policy for a flexible manufacturing cell feeding several production lines. IEE Transactions, v.16,n.4, p.355-362, december 1984.
- SEIFODDINI, Hamid. Machine-component group analysis versus the similarity coefficient method in cellular manufacturing applications. Computers Ind. Engng., v.18, n.3, p.333-339, 1990.
- SEPEHRI, Mehran, RAFFISH, Norm. Developing and implementing control systems for repetitive manufacturing. Industrial Engineering, p. 34-46, june 1986.
- SHAFER, Scott M., KERN, Gary M. and WEI, Jerry C. A mathematical programming approach for dealing with exceptional elements in cellular manufacturing. Int. J. Prod. Res., v.30, n.5, p.1029-1036, 1992.
- SHAW, Michael J. Knowledge- based scheduling in flexible manufacturing systems: An integration of pattern -directed inference and heuristic search. Int. Prod. Res., v.26, n.5, p. 821-844, 1988.
- SHAW, Michael J., WHINSTON, Andrew B. An artificial intelligence approach to the scheduling of flexible manufacturing systems. IIE Transactions, v.21, n.2, p.179-182, june 1989.
- SHEFFI, Yosef. Urban Transportation Networks : Equilibrium Analysis with Mathematical Programming. Prentice-Hall, 1985.
- SHTUB, Avraham. Modelling group technology cell formation as a generalized assignment problem. Int. J. Prod. Res., v.27, n.5, p.775-782, 1989.

- SHUNK, Dan L., FILLEY, Richard D. Systems integration series, part 1: Systems integration's challenges demand a new breed of industrial engineer. Industrial Engineering, v.18, n.5, p.65-67, may 1986.
- SLATTER, R. R. A human-centred approach to the design of advanced manufacturing systems. In: Annals of the CIRP, v.38, n.1, p.461-464, 1989.
- SPUR,G., MERTINS, K. Strategy- based interactive production control for flexible automted systems. In: Annals of CIRP, 1986.
- SSEMAKULA, Mukasa E., RANGACHAR, Ramesh M. The prospects of process sequence optimization in CAPP systems. Computers Ind. Engng., v.16, n.1, p. 162-170, 1989.
- STADTLER, Hartmut. Hierarchical production planning turning aggregate planning with sequencing and scheduling. Berlin: Spriger-Verlag, [199-].
- STECKE,K.E.. Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems. Management Science, v.29, n.3, p.273-288, 1983.
- STEMMER, Gaspar Erich. Tecnologia de grupo e células de fabricação. Revista de Sistemas e Equipamentos de Movimentação e Armazenagem de Materiais, p. 61-68, jan/fev 1983.
- TEICHOLZ, Eric. Computer integrated manufacturing. Datamation, v.30, n.3, p.169-174, mar. 1984.
- TEIXEIRA, Jorge Pamies , MACHADO, Virgílio Cruz. Novos modelos de classificação na tecnologia de grupo. Portugal: (Secção autonoma de Engenharia Mecânica), [199-].
- TEIXEIRA, Jorge Pamies. A tecnologia de grupo num ambiente integrado de CAD/CAM. Santiago do Chile, 1988 (documento apresentado no workshop do projeto CYTED. D, robótica y automatización aunzada).
- TESTA, D. et al. A computer based control system for flexible assembly lines. In: Annals of CIRP, 1986.

- TOM, Messman. Design considerations when integrating system modules Fall Industrial Engineering Conference Proceedings, p.307-316, 1981
- TSATSOULIS, Costas, KASHYAP, R. L. A system for knowledge-based process planning. Artificial intelligrnce in Engineering, v.3, n.2, p.61-75, 1988.
- VAKHARIA, Asoo J., WEMMERLOV Urban. Designing a cellular manufacturing systems: a materials flow approach based on operation sequences. III Transactions, v.22, n.1, p.84-96, march 1990.
- VENK, Sridharan, GOVIND, Rakesh. An expert system approach to optimization of the centerless grinding process. In: Annals of the CIRP, v.39, n.1, p.489-492, 1990.
- VESTER, John, VENCKUS, John. Systems integration series, part 9: Systems integration and the material spine. Industrial Engineering, p.48-52, jan 1987.
- VILA FILHO, Eduardo et al. Fabricação em familia de peças: um novo conceito em sistemas de produção. Máquinas e Ferramenta, p.36-40, agosto 1982 (primeira parte do artigo).
- VILA FILHO, Eduardo. Computer-aided group technology part family formation based on pattern recognition techniques.[S.L.], 1987. Thesis - Industrial Engineering.
- WANG, Hau-Pin, CHANG, Heng. Automated classification and coding based on extracted surface features in a CAD data. The Interntional Journal of Advanced Manufacturing technology, v.2, n.1, p.25-38, 1987.
- WANG, Hsu- Pin Ben. A layered architecture for manufacturing operation planning. Computers Ind. Engng., v.14, n.2, p.201-210, 1988.
- WARNECK, H. J. et al. Conclusions from simulation of small and large FMS projects. In: Annals of CIRP, 1986.
- WARNECK, H. J. FMS- research viewpoint. In: Annals of CIRP, 1986.

- WASSENHOVE, Luk N. Van, VANDERHENST, Pierter. Planning production in a bottleneck department. European Journal of Operational Research, v.12, p.127-137, 1983.
- WEI, Jerry C., KERN Gary M. Commonality analysis: a linear cell clustering algorithm for group technology. Int. J. Prod. Res., v.27, n.12, p.2053-2062, 1989.
- WEILL, P. Manufacturing information systems for FMS. In: Annals of CIRP, 1986.
- WEILL, R. et al. Survey of computer- aided process planning systems. In: Annals of the CIRP, v.31, n.3, p. 539-551, 1982.
- WELTER, William. Looking at the human side of manufacturing strategy for an effective CIMS. Industrial Engineering, p.18-21, june 1986.
- WEMMERLOV, U., HYER, Nancy L. Cellular manufacturing in the U.S. industry: a survey of users. Int. J. Prod. Res., v.27, n.9, p.1511-1530, 1989.
- WEMMERLOV, U., HYER, Nancy L. Research issues in cellular manufacturing. Int. J. Prod. Res., v.25, n.3, p.413-431, 1987.
- WILDE, Douglas J., BEIGHTLER, Charles S. Foundations of Optimization. Prentice-Hall, 1967.
- WILLIAMS, Jeffrey, NOVAK, Robert S. Aligning CIM strategies to different markets. Long Range Planning, v.23, p. 126-135, 1990.
- YAMASHINA, H.,et al. Operation of the flow type FMS. In: Annals of CIRP, 1986.
- ZELENOVIC, Dragutin M., TESIC, Zdravko M. Period batch control and group technology. Int. J. Prod. Res., v.26, n.3, p.539-552, 1988.

ANEXOS

GENETIC ALGORITHM FOR CELL FORMATION**GACELL - 1.0****MENU**

- 1 - CRIAR UM NOVO ARQUIVO
- 2 - ALTERAR ARQUIVO EXISTENTE
- 3 - RODAR ARQUIVO
- 4 - APAGAR ARQUIVO
- 5 - SAIR DO PROGRAMA

Digite a opção:

Anexo I: Menu Principal do GACELL

GENETIC ALGORITHM FOR CELL FORMATION GACELL - 1.0		
ARQUIVOS EXISTENTES:		
Test.INP	V.INP	VV.INP
X.INP	XA.INP	XC100020.INP
XC10020.INP	XC10040.INP	XC10060.INP
XC200020.INP	XC200022.INP	XC200023.INP
XC50020.INP	XC50010.INP	XC70.INP
XC90.INP	XC902.INP	XC902020.INP
XC904.INP	XC902080.INP	XC906.INP
VVV.INP	V.INP	VV.INP
Z.INP	XY.INP	XY100020.INP
XC20020.INP	XY10040.INP	XY10060.INP
XC400020.INP	XY200022.INP	XY200023.INP
XC20020.INP	XY50010.INP	XY70.INP
XZ90.INP	XY902.INP	XY902020.INP
XZ904.INP	XY902080.INP	XY906.INP
Entre com o nome do arquivo de dados: —————→		

Anexo II : Janela com Lista de Arquivos

GENETIC ALGORITHM FOR CELL FORMATION GACELL - 1.0
<p>ENTRE COM A OPÇÃO DESEJADA DE ALTERAÇÃO DE DADOS:</p> <p>1 - DIGITAR APENAS OS VALORES DO ALGORITMOGENÉTICO</p> <p>2 - DIGITAR TODOS OS VALORES</p> <p>OPÇÃO:</p>

Anexo III : Janela de Opções do *GACELL*

GENETIC ALGORITHM FOR CELL FORMATION GACELL - VERSÃO 1.0	
Digite os valores do GACELL :	
Qual o tamanho da população? (digite um núm. par entre 4 e 60)	20
Qual o número de peças? (digite um número entre 2 e 20)	15
Qual o número de máquinas? (digite um número entre 2 e 10)	10
Qual o número de grupos? (digite um número entre 2 e 5)	3
Qual o número máximo de tipos de peças por grupo? (digite um número entre 2 e 10)	5
Qual o número máximo de máquinas por grupo? (digite um valor entre 2 e 5)	5
Qual o número máximo de gerações? (digite um número entre 10 e 10000)	2000
Qual a probabilidade de cruzamento? (digite um número entre 4 e 100)	90.00
Qual a probabilidade de mutação? (digite um valor entre 0.1% e 10%)	1.00
Qual a dispersão máxima entre os indivíduos? (digite um número entre 0 e 10)	1.00
Qual o limite inferior de utilização do grupo? (digite um número entre 4 e 100)	30.00
Qual o limite superior de utilização do grupo? (digite um número entre 4 e 100)	95.00
Qual o custo ótimo esperado? (digite um número maior que 3)	75000.00
Qual a tolerância do custo ótimo esperado? (digite um número entre 0 e 10)	2.00
Qual o tipo de relatório de saída? (digite A=Arquivo ou T=Tela ou Q=Quick A=Arq)	

Anexo IV : Janela para Alteração dos Parâmetros do **GACELL**

**GENETIC ALGORITHM FOR CELL FORMATION
GACELL - VERSÃO 1.0**

Digite os parâmetros do GACELL

Qual o número máximo de planos da peça "1"? (digite um número entre 1 e 3)	2
Qual o número máximo de planos da peça "2"? (digite um número entre 1 e 3)	2
Qual o número máximo de planos da peça "3"? (digite um número entre 1 e 3)	3
Qual o número máximo de planos da peça "4"? (digite um número entre 1 e 3)	1
Qual o número máximo de planos da peça "5"? (digite um número entre 1 e 3)	2
Qual o número máximo de planos da peça "6"? (digite um número entre 1 e 3)	1
Qual o número máximo de planos da peça "7"? (digite um número entre 1 e 3)	1
Qual a quantidade de peças do tipo "1" devem ser produzidas? (digite um número maior ou igual a 1)	10
Qual a quantidade de peças do tipo "2" devem ser produzidas? (digite um número maior ou igual a 1)	12
Qual a quantidade de peças do tipo "3" devem ser produzidas? (digite um número maior ou igual a 1)	15
Qual a quantidade de peças do tipo "4" devem ser produzidas? (digite um número maior ou igual a 1)	5
Qual a quantidade de peças do tipo "5" devem ser produzidas? (digite um número maior ou igual a 1)	9
Qual a quantidade de peças do tipo "6" devem ser produzidas? (digite um número maior ou igual a 1)	8
Qual a quantidade de peças do tipo "7" devem ser produzidas? (digite um número maior ou igual a 1)	10

Anexo IV (continuação) : Janela para Alteração dos Parâmetros do **GACELL**

**GENETIC ALGORITHM FOR CELL
FORMATION GACELL - VERSÃO 1.0**

Digite os parâmetros do GACELL:

Qual o número de máquinas disponíveis do tipo "1" ? (digite um número entre 1 e 2)	3
Qual o número de máquinas disponíveis do tipo "2" ? (digite um número entre 1 e 2)	3
Qual o número de máquinas disponíveis do tipo "3" ? (digite um número entre 1 e 2)	3
Qual o número de máquinas disponíveis do tipo "4" ? (digite um número entre 1 e 2)	2
Qual o número de máquinas disponíveis do tipo "5" ? (digite um número entre 1 e 2)	2
Qual o número de máquinas disponíveis do tipo "6" ? (digite um número entre 1 e 2)	2
Qual o número de máquinas disponíveis do tipo "7" ? (digite um número entre 1 e 2)	3
Qual o número de ferramentas da máquina do tipo "1"? (digite um número maior ou igual a 1)	6
Qual o número de ferramentas da máquina do tipo "2"? (digite um número maior ou igual a 1)	6
Qual o número de ferramentas da máquina do tipo "3"? (digite um número maior ou igual a 1)	5
Qual o número de ferramentas da máquina do tipo "4"? (digite um número maior ou igual a 1)	6
Qual o número de ferramentas da máquina do tipo "5"? (digite um número maior ou igual a 1)	7
Qual o número de ferramentas da máquina do tipo "6"? (digite um número maior ou igual a 1)	6
Qual o número de ferramentas da máquina do tipo "7"? (digite um número maior ou igual a 1)	6

Anexo IV (continuação) : Janela para Alteração dos Parâmetros do **GACELL**

GENETIC ALGORITHM FOR CELL FORMATION GACELL - VERSÃO 1.0	
Digite os parâmetros do GACELL	
Qual o custo de capital da máquina do tipo "1"? (digite um número maior que 0)	100.00
Qual o custo de capital da máquina do tipo "2"? (digite um número maior que 0)	150.00
Qual o custo de capital da máquina do tipo "3"? (digite um número maior que 0)	200.00
Qual o custo de capital da máquina do tipo "4"? (digite um número maior que 0)	250.00
Qual o custo de capital da máquina do tipo "5"? (digite um número maior que 0)	300.00
Qual o custo de capital da máquina do tipo "6"? (digite um número maior que 0)	100.00
Qual o custo de capital da máquina do tipo "7"? (digite um número maior que 0)	150.00
Qual o tempo máximo de disponibilidade da máq. do tipo "1"? (digite o número de horas maior que 0)	27000.00
Qual o tempo máximo de disponibilidade da máq. do tipo "2"? (digite o número de horas maior que 0)	27000.00
Qual o tempo máximo de disponibilidade da máq. do tipo "3"? (digite o número de horas maior que 0)	27000.00
Qual o tempo máximo de disponibilidade da máq. do tipo "4"? (digite o número de horas maior que 0)	27000.00
Qual o tempo máximo de disponibilidade da máq. do tipo "5"? (digite o número de horas maior que 0)	27000.00
Qual o tempo máximo de disponibilidade da máq. do tipo "6"? (digite o número de horas maior que 0)	27000.00
Qual o tempo máximo de disponibilidade da máq. do tipo "7"? (digite o número de horas maior que 0)	27000.00

Anexo IV (continuação) : Janela para Alteração dos Parâmetros do **GACELL**

GENETIC ALGORITHM FOR CELL FORMATION GACELL - VERSÃO 1.0	
Dados da peça "1" com o plano "1"	
Qual a 1º máquina usada na peça "1" com o plano "1" ? (digite um número entre 1 e 10 ou 0 para encerrar)	2
Qual a 1º fer. da máq. "2" usada na peça "1" com o plano "1" ? (digite um número entre 1 e 6)	2
Qual o 1º tempo da fer. "2" usada na peça "1" com o plano "1" ? (digite um valor em minutos maior que 0)	1.00
Qual o 1º custo da ferramenta "2" usada na peça "1" com o plano "1" ? (digite um valor em dolares maior que 0)	4.50
Qual a 2º máquina usada na peça "1" com o plano "1" ? (digite um número entre 1 e 10 ou 0 para encerrar)	4
Qual a 2º ferramenta da máq. "4" usada na peça "1" com o plano "1" ? (digite um número entre 1 e 6)	1
Qual o 2º tempo da fer. "1" usada na peça "1" com o plano "1" ? (digite um valor em minutos maior que 0)	81.94
Qual o 2º custo da fer. "1" usada na peça "1" com o plano "1" ? (digite um valor em dolares maior que 0)	7.24
Qual a 3º máquina usada na peça "1" com o plano "1" ? (digite um número entre 1 e 10 ou 0 para encerrar)	4

Anexo IV (continuação) : Janela para Alteração dos Parâmetros do **GACELL**

GENETIC ALGORITHM FOR CELL FORMATION GACELL - VERSÃO 1.0	
Dados da peça "1" com o plano "1" (continuação)	
Qual a 7º máquina usada na peça "1" com o plano "1" ? (digite um número entre 1 e 10 ou 0 para encerrar)	8
Qual a 7º ferramenta da máq. "8" usada na peça "1" com o plano "1" ? (digite um número entre 1 e 6)	2
Qual o 7º tempo da fer. "2" usada na peça "1" com o plano "1" ? (digite um valor em minutos maior que 0)	7.63
Qual o 7º custo da fer. "2" usada na peça "1" com o plano "1" ? (digite um valor em dolares maior que 0)	17.45
Qual a 8º máquina usada na peça "1" com o plano "1" ? (digite um número entre 1 e 10 ou 0 para encerrar)	8
Qual a 8º ferramenta da máq. "8" usada na peça "1" com o plano "1" ? (digite um número entre 1 e 6)	3
Qual o 8º tempo da ferr. "3" usada na peça "1" com o plano "1" ? (digite um valor em minutos maior que 0)	6.61
Qual o 8º custo da ferr. "3" usada na peça "1" com o plano "1" ? (digite um valor em dolares maior que 0)	6.69
Qual a 9º máquina usada na peça "1" com o plano "1" ? (digite um número entre 1 e 10 ou 0 para encerrar)	9
Qual a 9º ferramenta da máq. "9" usada na peça "1" com o plano "1" ? (digite um número entre 1 e 4)	1

Anexo IV (continuação) : Janela para alteração dos parâmetros do *GACELL*

GENETIC ALGORITHM FOR CELL FORMATION GACELL - VERSÃO 1.0	
Dados da peça "1" com o plano "2"	
Qual a 1º máquina usada na peça "1" com o plano "2" ? (digite um número entre 1 e 10 ou 0 para encerrar)	1
Qual a 1º ferramenta da máq. "1" usada na peça "1" com o plano "2" ? (digite um número entre 1 e 6)	1
Qual o 1º tempo da ferr. "1" usada na peça "1" com o plano "2"? (digite um valor em minutos maior que 0)	66.58
Qual o 1º custo da ferramenta "1" usada na peça "1" com o plano "2"? (digite um valor em dolares maior que 0)	17.51
Qual a 2º máquina usada na peça "1" com o plano "2" ? (digite um número entre 1 e 10 ou 0 para encerrar)	4
Qual a 2º ferramenta da máq. "4" usada na peça "1" com o plano "2" ? (digite um número entre 1 e 6)	1
Qual o 2º tempo da ferr. "1" usada na peça "1" com o plano "2"? (digite um valor em minutos maior que 0)	66.49
Qual o 2º custo da ferr. "1" usada na peça "1" com o plano "2"? (digite um valor em dolares maior que 0)	6.91
Qual a 3º máquina usada na peça "1" com o plano "2" ? (digite um número entre 1 e 10 ou 0 para encerrar)	4
Qual a 3º ferramenta da máq. "4" usada na peça "1" com o plano "2" ? (digite um número entre 1 e 6)	2

Anexo IV (continuação) : Janela para Alteração dos Parâmetros do **GACELL**

GENETIC ALGORITHM FOR CELL FORMATION GACELL - VERSÃO 1.0	
Dados da peça "2" com o plano "1"	
Qual a 1º máquina usada na peça "2" com o plano "1" ? (digite um número entre 1 e 10 ou 0 para encerrar)	1
Qual a 1º ferramenta da máq. "1" usada na peça "2" com o plano "1" ? (digite um número entre 1 e 6)	2
Qual o 1º tempo da ferr. "2" usada na peça "2" com o plano "1"? (digite um valor em minutos maior que 0)	14.26
Qual o 1º custo da ferramenta "2" usada na peça "2" com o plano "1"? (digite um valor em dolares maior que 0)	22.50
Qual a 2º máquina usada na peça "2" com o plano "1" ? (digite um número entre 1 e 10 ou 0 para encerrar)	1
Qual a 2º ferramenta da máq. "1" usada na peça "2" com o plano "1" ? (digite um número entre 1 e 6)	3
Qual o 2º tempo da ferr. "3" usada na peça "2" com o plano "1"? (digite um valor em minutos maior que 0)	46.08
Qual o 2º custo da ferr. "3" usada na peça "2" com o plano "1"? (digite um valor em dolares maior que 0)	4.35
Qual a 3º máquina usada na peça "2" com o plano "1" ? (digite um número entre 1 e 10 ou 0 para encerrar)	1
Qual a 3º ferramenta da máq. "1" usada na peça "2" com o plano "1" ? (digite um número entre 1 e 6)	4

Anexo IV (continuação) : Janela para Alteração dos Parâmetros do **GACELL**

GENETIC ALGORITHM FOR CELL FORMATIONGACELL - VERSÃO 1.0		
ARQUIVOS EXISTENTES:		
Test.INP	V.INP	VV.INP
X.INP	XA.INP	XC10020.INP
XC10020.INP	XC10040.INP	XC10060.INP
XC200020.INP	XC200022.INP	XC200023.INP
XC50020.INP	XC50010.INP	XC70.INP
XC90.INP	XC902.INP	XC902020.INP
XC904.INP	XC902080.INP	XC906.INP
VVV.INP	V.INP	VV.INP
Z.INP	XY.INP	XY10020.INP
XC20020.INP	XY10040.INP	XY10040.INP
XC400020.INP	XY200022.INP	XY200023.INP
XC20020.INP	XY50010.INP	XY70.INP
XZ90.INP	XY902.INP	XY902020.INP
XZ904.INP	XY902080.INP	XY906.INP
Entre com o nome do arquivo a apagar:		XY906.INP
Vou apagar o arquivo:		Confirme (S/N)

Anexo IV: Janela para Alteração dos Parâmetros do **GACELL**