

MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

PROJETO DE FORMAÇÃO DE CÉLULAS DE MANUFATURA ATRAVÉS DA  
UTILIZAÇÃO DE ALGORITMOS GENÉTICOS

por

Rafael Crespo Izquierdo

Dissertação para obtenção do Título de  
Mestre em Engenharia

Porto Alegre, Maio de 2013

PROJETO DE FORMAÇÃO DE CÉLULAS DE MANUFATURA ATRAVÉS DA  
UTILIZAÇÃO DE ALGORITMOS GENÉTICOS

por

Rafael Crespo Izquierdo

Engenheiro Mecânico

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Mecânica, da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos necessários para a obtenção do Título de Mestre em Engenharia.

Mestre em Engenharia

Área de Concentração: Processos de Fabricação

Orientador: Prof. Dr. Flávio José Lorini

Co-orientador: Prof. Dr. Herbert Martins Gomes

Comissão de Avaliação:

Prof. Dr. Heraldo José de Amorim, PROMEC / UFRGS

Prof. Dr. Letícia Fleck Fadel Miguel, PROMEC / UFRGS

Prof. Dr. Giovana Savitri Pasa, PPGE/ UFRGS

Prof. Dr. Rogério José Marczak

Coordenador do PROMEC

Porto Alegre, 29 Maio de 2013.

Determinação, coragem e autoconfiança são fatores decisivos para o sucesso.  
Se estamos possuídos por uma inabalável determinação conseguiremos superá-los.  
Independentemente das circunstâncias, devemos ser sempre humildes, recatados e despidos de orgulho.

Dalai Lama

## AGRADECIMENTOS

A Deus, por me proporcionar chegar até esta etapa da minha vida.

A minha querida mãe e amiga, Sabrina Pereira de Abreu, por todo o amor, apoio, carinho e compreensão; Mãe, a tua presença é fundamental para minha caminhada até hoje. Muito obrigado, eu te amo.

Ao meu querido pai, Carlos Crespo Izquierdo, por toda sua atenção e carinho, sempre compreendendo minha ausência durante este período. Muchas gracias, Papa.

A minha amada família que sempre me acompanhou e ajudou em todas as etapas da minha vida. Muito obrigado: meu primo-irmão, Luis Fernando; minhas avós, Maria e Ricarda, e a todos que torceram por mim.

A minha companheira Cássia Pederiva de Oliveira, por toda sua dedicação e compreensão durante a realização desta dissertação.

Ao meu orientador, professor Dr. Flávio José Lorini, por ter se envolvido neste trabalho, orientando cada etapa de execução desta pesquisa com competência, companheirismo e dedicação.

Ao meu coorientador, professor Dr. Herbert Martins Gomes, por ter abraçado a nossa causa, minha e do professor Lorini, sempre apoiando nos momentos necessários.

Aos professores do PROMEC, por terem participado da formação do meu conhecimento, obrigado.

Aos funcionários do PROMEC, por todo apoio.

Aos meus queridos amigos pessoais e aos amigos do PROMEC por todo apoio, carinho, compreensão e amizade.

Ao CNPq, por ter me proporcionado realizar o mestrado com o auxílio da bolsa.

Rafael Crespo Izquierdo

## RESUMO

Nos modernos sistemas produtivos exige-se crescente aumento na eficiência global, o que implica em otimização nas etapas de processo. Neste contexto, o arranjo de um layout industrial é um conceito fundamental para a eficácia de um sistema, que deve ser considerada na sua concepção. Em um ambiente industrial podem-se conceber diversos tipos de layout, de acordo com a diversidade de itens e do modo de produção adotado, da demanda de mercado. O presente trabalho aborda o projeto de layout do tipo celular, uma modalidade de arranjo utilizado na engenharia industrial que permite atender diversidade de produção compatível com certa flexibilidade operacional. Dentre as diversas técnicas e abordagens aplicadas para a formação de células de manufatura, adota-se especificamente uma aplicação de Algoritmo Genético, implementada através de uma interface em *Matlab*, onde são abordados tópicos como: a geração da população inicial, a codificação do cromossomo, a função objetivo, as restrições do problema, os operadores de cruzamento e mutação e a confiabilidade do algoritmo proposto. Dos resultados obtidos, constata-se que os Algoritmos Genéticos são ferramentas confiáveis para a otimização de sistemas de fabricação, obtendo-se respostas que cumprem os requisitos básicos de projeto de uma célula, além de apresentar valores de convergência na geração de grupos de máquinas e famílias de peças.

**Palavras-chave:** Algoritmos Genéticos; Células de manufatura; Layout industrial.

## **ABSTRACT**

In modern production systems it requires increasing the overall efficiency, which results in the optimization process steps. In this context, the arrangement of an industrial layout is a fundamental concept to the effectiveness of a system that must be considered in its design. An industrial environment can be designed with layouts of several types, according to the diversity of items, adopted production mode and market demand. This paper discusses the design layout of the cellular type, an arrangement used in industrial engineering that allows diversity to meet production compatible with operational flexibility. Among the various techniques and approaches applied to the formation of manufacturing cells, specifically here it is adopted an application of genetic algorithm, implemented using an interface in Matlab, where several topics are covered, namely: the generation of the initial population, the chromosome coding, operators of Crossover, mutation and reliability of the proposed algorithm. The obtained results show that the Genetic Algorithms are reliable tools for the optimization of manufacturing systems, obtaining responses that meet the basic requirements of a cell design, and presenting convergence values in the generation of machine groups and family of parts.

**Keywords:** Genetic Algorithms; Cells manufacturing; Industrial Layout.

## ÍNDICE

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>13</b>
1.1	Objetivo do Trabalho.....	14
1.2	Objetivos específicos do Trabalho.....	14
1.3	Justificativa para a realização do Trabalho.....	14
1.4	Organização do Trabalho.....	15
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA .....</b>	<b>16</b>
2.1	Projeto e Planejamento de um Layout.....	16
2.2	Tipos de Layout.....	17
2.2.1	Layout Posicional (Posição Fixa) .....	18
2.2.2	Layout em Linha (por produto) .....	19
2.2.3	Layout Funcional (por processo) .....	20
2.2.4	Layout Celular – <i>Cellular Layout</i> .....	22
2.3	Técnicas de Formação de Células.....	24
2.3.1	Tecnologia de Grupo .....	24
2.3.2	Inspeção Visual.....	25
2.3.3	Classificação de peças e codificação.....	26
2.3.4	Análise do Fluxo de Produção (PFA).....	28
2.3.5	Algoritmos e Técnicas de Agrupamento.....	29
2.3.5.1	Programação Matemática.....	30
2.3.5.2	Técnicas de Agrupamento Hierárquico.....	30
2.3.5.3	Técnicas baseadas em arranjo.....	30
2.3.5.4	Técnicas de Agrupamento Não - Hierárquico (Técnicas Heurísticas).....	39
2.4	Algoritmos Genéticos - Caracterização.....	40
2.4.1	Introdução ao Algoritmos Genéticos.....	40
2.4.2	Terminologia dos Algoritmos Genéticos.....	41
2.4.3	Estrutura dos Algoritmos Genéticos.....	44
2.4.4	Características operacionais dos Algoritmos Genéticos.....	45
2.4.5	Operadores Genéticos.....	46
2.4.5.1	Codificação do Cromossomo.....	47
2.4.5.2	Função de Avaliação ou Objetivo.....	47
2.4.5.3	Seleção de indivíduos.....	48
2.4.5.4	Operador de Cruzamento ( <i>Crossover</i> ).....	51
2.4.5.5	Mutação.....	53
<b>3</b>	<b>IMPLEMENTAÇÃO DO ALGORITMO NA FORMAÇÃO DE CÉLULAS.....</b>	<b>54</b>
3.1	Introdução.....	54
3.2	<i>matlab</i> : Aplicação dos Algoritmos Genéticos.....	55
3.3	Geração da população inicial na formação de células.....	56
3.3.1	Tipo de população.....	56
3.3.2	Tamanho da população.....	58
3.4	Função Objetivo ( <i>fitness</i> ).....	59
3.4.1	Cálculo da Função F1.....	60
3.4.2	Cálculo da Função F2.....	61
3.4.3	Restrições na formação da população inicial e função objetivo.....	62
3.5	Escalonamento da função objetivo e Critérios de Seleção.....	63

3.6	Operadores Genéticos.....	64
3.7	Critérios de parada e convergência.....	64
3.8	Monitoramento e análise dos resultados.....	65
3.9	Funcionamento do Algoritmo proposto na formação de células.....	66
3.9.1	Características funcionais do algoritmo.....	66
3.9.2	Funcionamento da interface gráfica.....	70
<b>4</b>	<b>RESULTADOS: EXEMPLOS DE APLICAÇÃO DO APLICATIVO PROPOSTO.....</b>	<b>73</b>
4.1	Análise da matriz A.....	73
4.2	Análise da matriz B.....	75
4.3	Análise da matriz C.....	78
4.4	Análise comparativa através da variação do número de células.....	81
4.5	Análise comparativa com a literatura.....	82
4.6	Confiabilidade do aplicativo proposto.....	83
<b>5</b>	<b>CONCLUSÕES .....</b>	<b>85</b>
<b>6</b>	<b>SUGESTÕES DE TRABALHOS FUTUROS.....</b>	<b>87</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>88</b>



## LISTA DE FIGURAS

Figura 2.1	Relação entre variedade-volume na seleção de um arranjo Físico.....	18
Figura 2.2	Disposição de um Layout posicional ou de posição fixa.....	19
Figura 2.3	Disposição de um Layout por produto ou em linha.....	20
Figura 2.4	Disposição de um Layout por processo ou funcional.....	21
Figura 2.5	Disposição de um Layout Celular.....	23
Figura 2.6	Representação do método de Inspeção Visual.....	26
Figura 2.7	Representação básica do sistema Optiz.....	27
Figura 2.8	Representação de uma Matriz de incidência peça-máquina.....	29
Figura 2.9	Matriz de incidência peça-máquina: Ordenamento das linhas.....	31
Figura 2.10	Matriz de incidência peça-máquina: Ordenamento das colunas.....	32
Figura 2.11	Matriz de incidência peça-máquina: Solução ROC.....	32
Figura 2.12	Matriz de incidência peça-máquina: Método ROC 2.....	34
Figura 2.13	Matriz de incidência peça-máquina: Ordenamento das linhas.....	34
Figura 2.14	Matriz de incidência peça-máquina: Ordenamento das linhas e colunas.....	35
Figura 2.15	Rearranjo das linhas e colunas da matriz - Etapa 1.....	36
Figura 2.16	Rearranjo das linhas e colunas da matriz - Etapa 2.....	36
Figura 2.17	Matriz de incidência peça-máquina: Solução DCA.....	37
Figura 2.18	Representação da matriz com os passos 1-3: Método CIA.....	38
Figura 2.19	Matriz de incidência peça-máquina: Solução CIA.....	38
Figura 2.20	Representação do funcionamento de um Algoritmo Genético.....	45
Figura 2.21	Representação do funcionamento da Função Objetivo.....	48
Figura 2.22	Seleção via método da Roleta.....	49
Figura 2.23	Crossover de um ponto na representação binária.....	51
Figura 2.24	Crossover de dois pontos na representação binária.....	52
Figura 2.25	Crossover uniforme apresentado por Guerra, 2008.....	52
Figura 2.26	Mutação: a) bit a bit b) Inversão dos bits.....	53
Figura 3.1	Fluxograma de implementação do Algoritmo.....	54
Figura 3.2	Representação dos AG por linhas de comando em linguagem Matlab	55
Figura 3.3	Interface principal de aplicação dos AG.....	55
Figura 3.4	Representação de um cromossomo em bits.....	56

Figura 3.5	Representação de um cromossomo em números reais.....	56
Figura 3.6	Representação de um cromossomo para formação de células.....	57
Figura 3.7	Algoritmo de conversão de valores binários para inteiros.....	57
Figura 3.8	Representação de um população composta por N indivíduos.....	58
Figura 3.9	Princípio de funcionamento da função F1.....	60
Figura 3.10	Requisitos para formação da população inicial.....	62
Figura 3.11	Algoritmo de restrição da população inicial.....	63
Figura 3.12	Princípio de funcionamento da ferramenta por escalonamento.....	64
Figura 3.13	Princípio de funcionamento do AG quando ao critério de parada por gerações.....	65
Figura 3.14	Valores monitorados pelo AG.....	66
Figura 3.15	Fluxograma de funcionamento do algoritmo proposto.....	67
Figura 3.16	Representação da chamada do toolbox por linha de comando.....	68
Figura 3.17	Representação da chamada do cálculo da função objetivo.....	69
Figura 3.18	Representação da chamada da sub-rotina para impressão.....	69
Figura 3.19	Interface do usuário para definição das variáveis.....	70
Figura 3.20	Interface principal do aplicativo proposto.....	71
Figura 3.21	Inserindo o arquivo de dados do usuário.....	71
Figura 3.22	Interface de monitoramento do AG .....	72
Figura 3.23	Interface de resultados do aplicativo .....	72
Figura 4.1	Matriz de incidência peça-máquina na sua forma inicial.....	73
Figura 4.2	Resultados obtidos relacionados a análise dos AG - Exemplo 1.....	74
Figura 4.3	Solução do problema proposto.....	75
Figura 4.4	Matriz de incidência peça-máquina na forma inicial.....	75
Figura 4.5	Comparativo da resposta obtida no trabalho analisado e o aplicativo....	77
Figura 4.6	Resultados obtidos relacionados a análise dos AG - Exemplo 2.....	77
Figura 4.7	Matriz de incidência peça-máquina na sua forma inicial.....	78
Figura 4.8	Resultados obtidos por Burbridge [Chan et. al.,1982].....	79
Figura 4.9	Resultados obtidos com o aplicativo proposto.....	79
Figura 4.10	Resultado proposto por Burbridge [Chan et al, 1982].....	80
Figura 4.11	Resultados obtidos relacionados a análise dos AG - Exemplo 3.....	80

## LISTA DE TABELAS

Tabela 4.1	Parâmetros dos AG para a otimização do sistema.....	73
Tabela 4.2	Parâmetros para a formação de células de manufatura.....	73
Tabela 4.3	Parâmetros dos AG para a otimização do sistema.....	75
Tabela 4.4	Parâmetros para a formação de células de manufatura.....	75
Tabela 4.5	Parâmetros dos AG para a otimização do sistema.....	77
Tabela 4.6	Parâmetros para a formação de células de manufatura.....	77
Tabela 4.7	Relação entre função objetivo e a variação do número de células	81
Tabela 4.8	Análise comparativa entre o aplicativo e a literatura.....	82

## LISTA DE SIGLAS E ABREVIATURAS

AG	Algoritmos Genéticos
PFA	Análise do Fluxo de produção
BEA	<i>Bond Energy Algorithm</i>
CIA	<i>Cluster Identification Algorithm</i>
DCA	<i>Direct Clustering Algorithm</i>
FC	<i>Fuzzy and Clustering</i>
GA	<i>Genetic Algorithm</i>
GRUCON	Grupo de Pesquisas e Treinamento em Comando Numérico e Automatização Industrial
GT	<i>Group Technology</i>
JIT	<i>Just in Time</i>
JSPMI	Sociedade Japonesa para Promoção de Máquinas Industriais
MODROC	<i>Modified Rank Order Clustering</i>
ME	Medida de efetividade
MICLASS	<i>Metal Institute Classification System</i>
OIR	Organização para Pesquisa Industrial
ROC	<i>Rank Order Clustering</i>
ROC 2	<i>Rank Order Clustering 2</i>
RN	Redes Neurais
SA	<i>Simulated Annealing</i>
STP	Sistema Toyota de Produção
SCC	Sistema de Codificação e Classificação
SLC	<i>Single Linkage Clustering</i>
TG	Tecnologia de Grupo
TS	<i>Tabu Search</i>
TNO	<i>The Netherlands Organization for Applied Scientific Research</i>
WIP	<i>Work in process</i>

## LISTA DE SÍMBOLOS

$t_{ij}$	Tempo de execução da peça $j$ na máquina $i$ , h/peça.
$N_j$	Demanda da peça $j$ em um dado período de tempo
$F$	Função objetivo
$F1$	Função: minimização do volume de movimentos entre as células
$F2$	Função: minimização da variação da carga total dentro da célula
$w_{ij}$	Carga de trabalho da máquina $i$ na peça $j$
$e_{ji}$	Matriz que relaciona se existe tempo de fabricação na máquina $i$ com peça $j$
$x_{il}$	Matriz que relaciona se a máquina $i$ pertence à célula $l$
$N$	Número de elementos da população (cromossomos).
$K$	Número de células a serem formadas

# 1 INTRODUÇÃO

As crescentes exigências de produtividade sobre os sistemas produtivos fazem com que uma análise no layout seja fundamental, pois o arranjo físico das máquinas em um ambiente fabril pode influenciar no fluxo de produção e, conseqüentemente, na eficiência e custo de um processo produtivo. No entanto, para alcançar a eficiência desejada em um processo, reduzindo os custos, também é necessário conhecer a demanda de peças utilizadas, com intuito de prever o fluxo de produção utilizado, visando sempre aperfeiçoar a condição existente.

Atualmente grande parte dos sistemas produtivos realizam a produção centrada nas necessidades dos clientes “*just in time*”(JIT), porque é uma maneira de não gerar estoques. Isto faz com que a utilização de layout celulares sejam mais apropriados para suprir este tipo de demanda. Em um ambiente industrial existem diversos tipos que layouts que são utilizados de acordo com o tipo de sistema produtivo, de modo a atender as demandas existentes no mercado com relação aos tipos de peças a serem produzidas. Entre os tipos de layout pode-se citar os seguintes: Linha, Funcional, Posicional e Celular. No caso do layout celular, trata-se de um arranjo de aplicação crescente na engenharia industrial, pois ele permite bons níveis de produção com uma flexibilidade operacional. Para seu projeto a literatura apresenta diversas abordagens de métodos que visam formar células através de técnicas específicas.

No projeto de uma célula, o essencial, como ponto de partida, consiste no agrupamento de peças em famílias para os respectivos grupos de máquinas, associadas a essas famílias.

Diversos métodos utilizados como ferramentas para essa finalidade trabalham com as informações relativas ao fluxo dos processos das peças atreladas as respectivas máquinas tabuladas na forma de uma matriz peça-máquina. Essa matriz, uma vez trabalhada por determinado algoritmo pode apresentar nas colunas próximas aquelas peças de fluxo semelhante, bem como nas linhas as máquinas associadas. Essa matriz diagonalizada representa uma combinação de proposta de células.

Para realizar a diagonalização da matriz de incidência peça-máquina, existem diversos métodos que podem ser utilizados, como ROC (Rank Order Clustering), Classificação e Codificação e os métodos Heurísticos. Neste trabalho estuda-se um método heurístico, que

está relacionado com a otimização de sistemas de manufatura através de processos iterativos, utilizando funções específicas e critérios de convergência.

Entre as diversas técnicas utilizadas nos métodos heurísticos podem-se citar aquelas associadas a Redes Neurais, Lógica Fuzzy, Agrupamentos Hierárquicos e Algoritmos Genéticos.

Neste trabalho estuda-se a implementação da formação de células de manufatura, utilizando os algoritmos genéticos, através da ferramenta do software *Matlab* chamada *gatool*.

### **1.1 Objetivo do Trabalho**

Propor a utilização de um Algoritmo Genético (AG) para a otimização de um sistema de fabricação, que consiste na formação de células de manufatura (layout celular) através dos recursos disponíveis da ferramenta *toolbox* do *Matlab*, visando comparar a implementação deste algoritmo com outras ferramentas e aplicações técnico-científicas existentes nesta área.

### **1.2 Objetivos específicos do Trabalho**

- Pesquisar e selecionar as variáveis disponíveis na ferramenta *toolbox* (AG) do *Matlab* para melhor convergência do algoritmo proposto.
- Estabelecer um comparativo entre o algoritmo genético implementado e as demais técnicas de formação de células de manufatura existentes.
- Desenvolver um aplicativo que integre os tópicos estudados e abordados neste trabalho, com o intuito de formar células de manufatura. Este aplicativo consiste na combinação de um Algoritmo, gerado através do *Matlab*, com uma interface construída também em linguagem *Matlab*, que permita realizar análises com relação à formação de células, utilizando este tipo de técnica de otimização.

### **1.3 Justificativa para a realização do trabalho**

Atualmente existem diversas pesquisas na área de projetos de layout, desde estudos de aplicação das técnicas convencionais aplicadas nos ambientes fabris, até a implementação de novas técnicas heurísticas, a fim de buscar resultados mais eficientes quanto à produtividade e custos em células de manufatura. Assim, a realização do presente trabalho se justifica em função de que o aplicativo proposto nesta pesquisa, permita abordar problemas utilizando um Algoritmo Genético (AG), com o suporte de um software convencional, como o *Matlab*, possibilitando comparar os dados obtidos utilizando os Algoritmos Genéticos com outras técnicas de otimização existentes.

#### 1.4 Organização do trabalho

Este trabalho está com o conteúdo disposto em 6 capítulos:

No capítulo 2 é apresentada a revisão bibliográfica utilizada para a fundamentação teórica, abordando os seguintes assuntos: conceitos de projeto de layout, conceitos de um layout celular, técnicas de formação de células de manufatura e os Algoritmos Genéticos (AG) e suas aplicações.

No capítulo 3 é discutida a metodologia para a implementação do AG na formação de células, de forma que aborda-se os seguintes conceitos: a estrutura do algoritmo, a interface gráfica, as definições dos parâmetros dos AG, e o funcionamento do *toolbox gatool* do *Matlab*.

No capítulo 4 realiza-se uma análise dos resultados obtidos pelo algoritmo proposto, onde são utilizados três exemplos para avaliar sua robustez, bem como comparar os resultados com a bibliografia proposta, visando verificar a confiabilidade da metodologia implementada.

No capítulo 5 apresentam-se as conclusões do trabalho, onde são analisados os conceitos, tais como: características do aplicativo proposto, parâmetros dos AG e suas influências, robustez e confiabilidade do algoritmo.

No capítulo 6 são apresentadas algumas sugestões de trabalhos futuros.



## 2 REVISÃO BIBLIOGRÁFICA

Uma revisão bibliográfica, como fundamentação teórica para o embasamento inerente ao desenvolvimento do trabalho é apresentada a seguir.

### 2.1 Projeto e planejamento de um Layout industrial

Quando se realiza algum tipo de modificação em um layout industrial, o custo para realizar estas alterações é relativamente alto, assim é extremamente importante fazer um planejamento do arranjo a ser utilizado antes de sua implementação. Isso faz com que as perdas associadas com tempo de movimentação de máquina e equipamentos sejam amenizadas, em função de que este planejamento inicial consiste em uma sequência lógica para futuras mudanças [Muther, 1973].

Segundo Slack *et. al.*, 1999, o arranjo físico ou layout de um processo produtivo está relacionado com a disposição física de todos os elementos necessários para a transformação de um determinado produto. Desta forma, o projeto de um layout industrial deve prever as instalações, as máquinas e a mão de obra utilizada. Então, se o layout de um determinado ambiente fabril encontra-se com sua disposição incorreta, pode acarretar certa descontinuidade no fluxo de produção com tempos de processamento longos, e por consequência, gerando custos elevados, devido à presença de estoques em processo. Com relação aos conceitos de implementação de um layout, Muther, 1973, complementa que para o planejamento das instalações de um arranjo físico devem-se considerar os seguintes itens:

- produto (material): relaciona-se com o que se deve produzir;
- quantidade (volume): associado com a quantidade que se deve produzir;
- roteiro: significa a sequência do processo de fabricação a ser considerada;
- serviços de suporte: associados com serviços de manutenção, alimentação, áreas de estoques, enfermaria, etc;
- tempo: relaciona-se com o período em que serão produzidos os itens.

A importância no planejamento e tomada de decisões de um layout é fundamental para a estratégia de uma empresa, pois com um projeto bem elaborado obtém-se desempenhos desejáveis, como tipos de arranjo físico que favorecem a flexibilidade das operações e a eficiência dos fluxos e uso dos recursos. No entanto, existem casos em que uma tomada de decisão no projeto pode gerar conflitos entre a obtenção da flexibilidade e eficiência,

considerando somente o projeto do arranjo físico. Assim, torna-se necessário adequar a decisão do arranjo físico de acordo com a estratégia mais competitiva da operação. [Corrêa e Corrêa, 2006].

De acordo com Shingo, 1996, em um ambiente industrial, a interface homem-máquina deve ser tratada separadamente, ou seja, o layout das máquinas deve-se relacionar com o fluxo de processamento de um produto, enquanto o fluxo de pessoas deve considerar os tempos de processamentos de um produto, projetando-se as operações com intuito de maximizar a eficiência do operador. Assim, observa-se que em um Sistema Toyota de Produção (STP) o balanceamento da carga dos trabalhadores busca ocupá-los intensamente e consolidar as ociosidades de capacidade num único operador, de modo que, posteriormente, ações de melhoria viabilizem que este seja realocado ou eliminado. Estes pontos citados evidenciam a importância de um estudo de otimização para formação de células de manufatura.

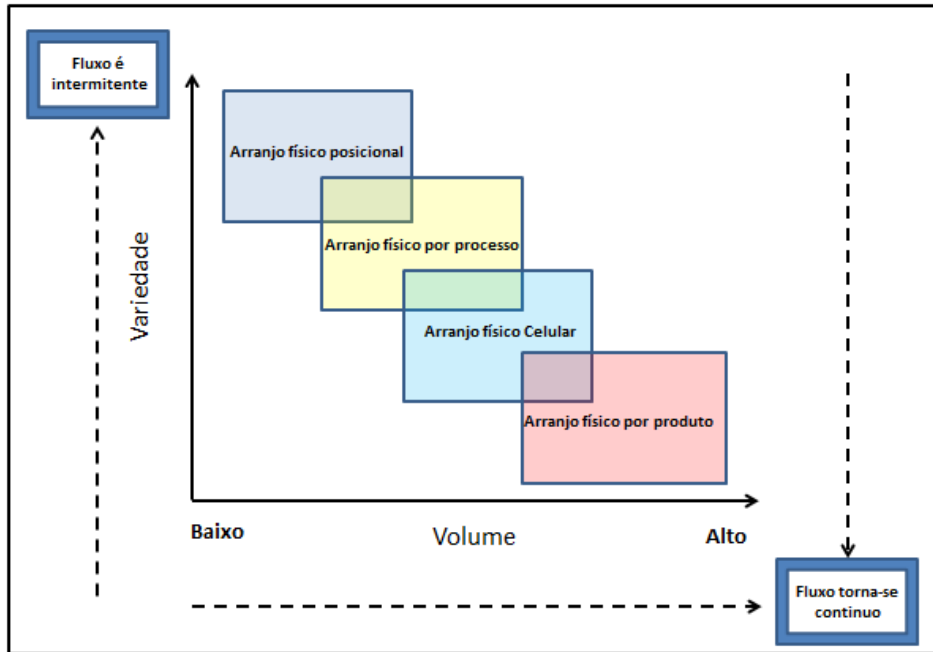
Segundo Lorenzatto *et. al.*, 2007, a otimização de um layout elimina uma série de perdas relacionadas a um determinado processo produtivo, como redução do tempo de mão de obra envolvida no transporte de materiais, aumento da produtividade, melhoria nos índices de qualidade e motivação dos funcionários.

De acordo com Black, 1998, um sistema de manufatura consiste em um conjunto integrado, de subsistemas interagindo entre si. Isto significa que otimizar partes do sistema, isto é, processos ou subsistemas, não necessariamente otimiza um sistema por inteiro.

## **2.2 Tipos de Layout**

Segundo Slack *et. al.*, 1999, o tipo básico de arranjo físico é a forma geral do arranjo de recursos produtivos, onde existem muitas maneiras diferentes de se arranjar estes recursos. A maioria dos layouts existentes em ambientes fabris estão relacionados com quatro tipos básicos de arranjos físicos, que são: layout Posicional (Posição Fixa), layout em linha (por produto), layout Funcional (por processo) e layout Celular.

De acordo com o autor, a escolha do tipo de layout a ser adotado pode ser definida através da relação entre a variedade de produtos a ser produzidos e o volume de produção, conforme ilustrado na figura 2.1.



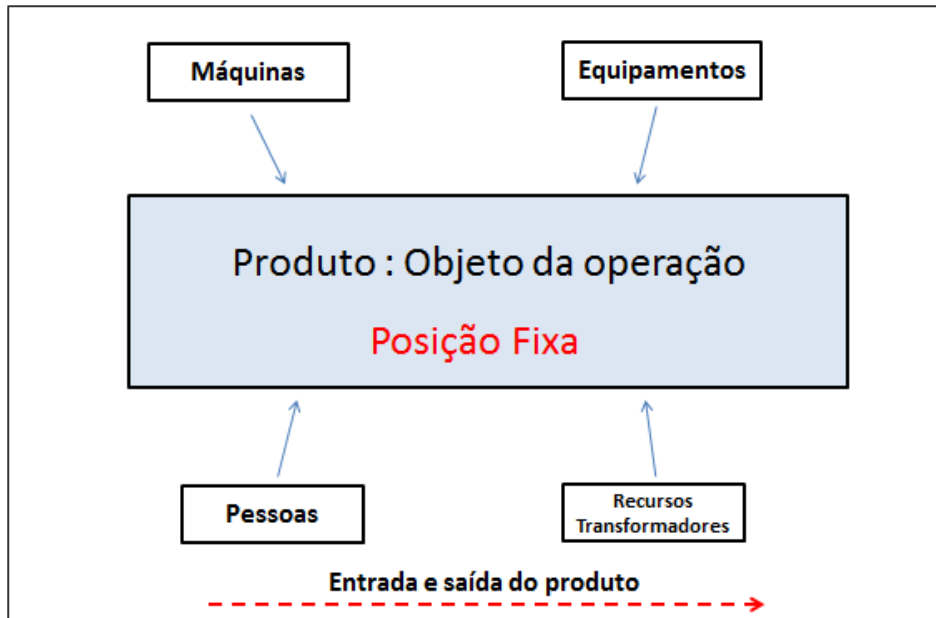
**Figura. 2.1- Relação entre variedade-volume na seleção de um arranjo Físico - adaptado de Slack *et. al.*, 1999**

Na figura 2.1 observa-se que para uma determinada faixa de variedade de produtos e volume de produção, as características dos arranjos físicos se sobrepõem, sendo necessária uma análise para definir qual o tipo de layout que se deve escolher, considerando as vantagens e desvantagens de cada arranjo físico. Esta relação entre variedade-volume também influencia no fluxo dos recursos transformados, isto é, quanto maior a variedade dos produtos e menor o volume de produção o fluxo vai se tornando intermitente. No entanto, quanto maior o volume de produção e menor a variedade dos produtos, mais contínuo o fluxo vai se tornando.

### 2.2.1 Layout Posicional (Posição Fixa)

De acordo com Slack *et. al.*, 1999, este tipo de layout se caracteriza por operações de montagem de forma que os materiais, pessoas e equipamentos, no caso os recursos transformadores, se deslocam até o objeto da operação, que também são conhecidos como os recursos transformados, conforme exemplificado na figura 2.2. Isto se deve em função de que o objeto da operação muitas vezes encontra-se estacionário por impossibilidade ou inconveniência de movê-lo nas etapas do processo de agregação de valor. [Corrêa e Corrêa, 2006].

Segundo Black, 1998, o arranjo físico posicional se caracteriza por não se produzir uma grande quantidade de itens finais, mas o tamanho do lote dos componentes pode variar de valores pequenos a muito grandes.



**Figura. 2.2 - Disposição de um Layout Posicional ou de posição fixa**

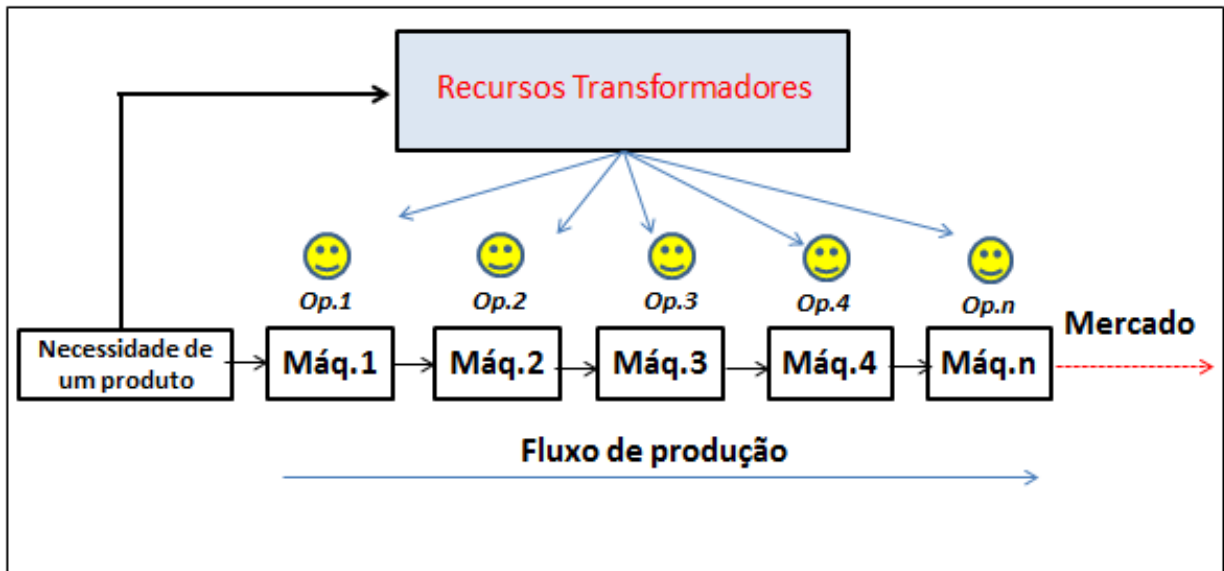
Entre as aplicações típicas para este tipo de layout, podem-se citar a fabricação de aviões, de navios, turbinas de grande porte, prensas de grande porte, pontes, locomotivas, etc.

### 2.2.2 Layout em Linha (por produto)

A característica principal do layout por produto é que os recursos utilizados para a transformação de um determinado produto, como, materiais, pessoas e equipamentos, são arranjados de forma adequada com o produto que está sendo produzido. Isto quer dizer que de acordo com um determinado produto existe uma sequência específica de máquinas e operações. [Slack *et. al.*, 1999], como ilustrado na figura 2.3. Lorini, 1993, aborda que esse layout se utiliza em processos industriais simples, linhas de montagem ou produção em massa, o que caracteriza a utilização de máquinas específicas e sistemas de transporte e manuseio dedicados.

Segundo Burbidge, 1983, este tipo de layout é difícil de se aplicar economicamente em processos de usinagem, a menos que a demanda necessária seja elevada. Isto se deve ao fato

de que natureza do trabalho de cada máquina em usinagem ser diferente, implicando em que os tempos de processamento não sejam iguais, podendo deixar máquinas de uma linha em espera, tornando-a desbalanceada.



**Figura. 2.3 - Disposição de um Layout por produto ou em linha**

Entre as aplicações para este tipo de layout, podem-se citar linhas de montagens de automóveis, industriais químicas, alimentícias, etc.

### 2.2.3 Layout Funcional (por processo)

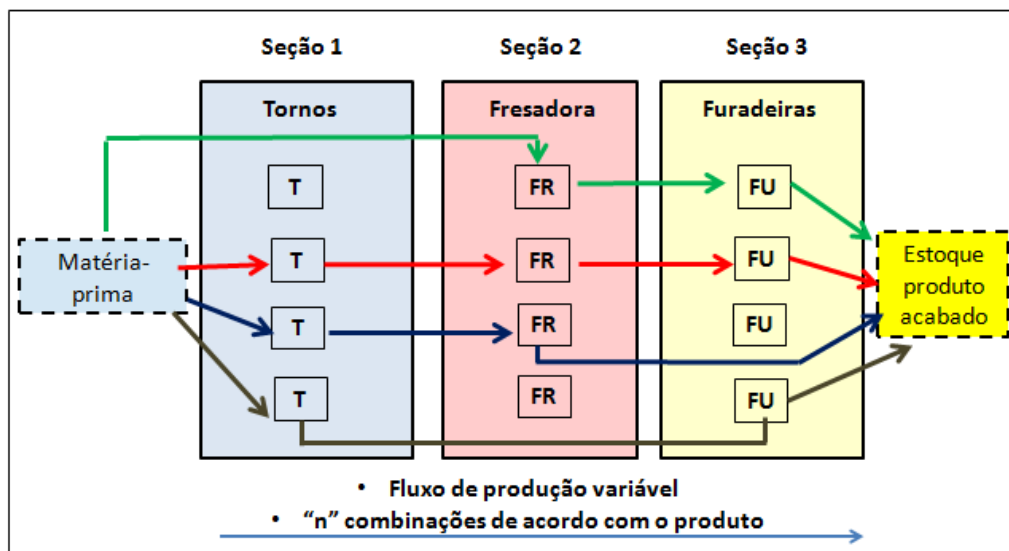
O conceito deste tipo de arranjo físico é que todos os diferentes tipos de equipamentos são agrupados de acordo com o processo. [Burbidge, 1983]. Ainda segundo Lorini, 1993, neste tipo de layout todas as máquinas de um mesmo tipo são dispostas juntas, formando um arranjo físico de especialização por processo, ou seja, máquinas que realizam o mesmo processo são agrupadas numa mesma área física. Este tipo de layout é o mais comum e encontrado em instalações industriais mais antigas.

Segundo Corrêa e Corrêa, 2006, este tipo de layout se caracteriza pelo fluxo variado de materiais entre as zonas agrupadas (setores). Assim, este tipo de arranjo físico é considerado bastante flexível, no entanto, se exige cuidados especiais quando os fluxos aumentam, pois pode ocorrer dos fluxos se cruzarem, diminuindo a eficiência e aumentando o tempo de atravessamento dos produtos. Black, 1998, ainda complementa que este tipo de

arranjo caracteriza-se na produção de uma grande variedade de produtos, que resulta em pequenos lotes de produção.

Uma das vantagens deste tipo de arranjo físico é que permite a especialização de supervisão e de mão de obra, considerando operadores semi-habilitados em equipamentos universais. Além disso, se os operadores são altamente treinados, este tipo de layout possibilita a transferência de mão de obra de uma máquina para outra. Outra vantagem do layout funcional é que permite uma melhor ocupação do equipamento, considerando um número pequeno de máquinas do mesmo tipo. No entanto, este tipo de layout possui algumas desvantagens, como dificultar um bom controle e supervisão. [Burbidge, 1983].

Na figura 2.4 representa-se o conceito de um layout funcional, onde máquinas encontram-se dispostas por tipo em um mesmo grupo, constituindo seções, de acordo com a funcionalidade específica.



**Figura. 2.4 - Disposição de um Layout por processo ou funcional**

Entre as aplicações típicas para este tipo de layout, podem-se citar linhas de usinagem, supermercados, hospitais e lojas de departamento, etc.

#### 2.2.4 Layout Celular

De acordo com Lorini, 1993, o layout celular se caracteriza pelas máquinas serem arranjadas em grupos de diversos tipos, com o objetivo de atender a produção, não mais de um determinado produto, mas sim de uma família de peças. Isto significa que este tipo de

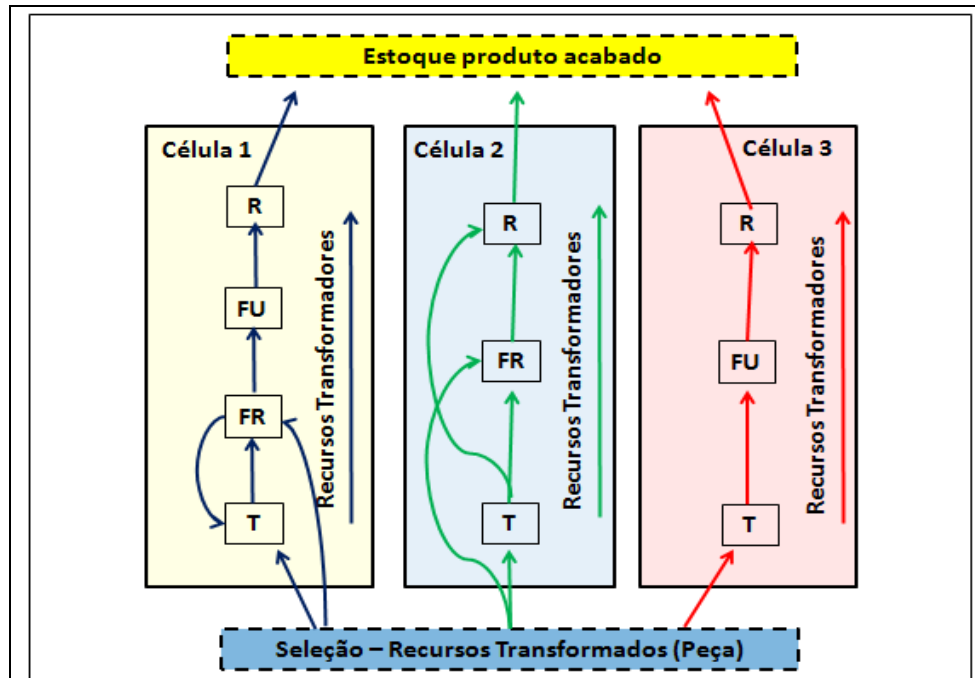
arranjo físico é aquele em que os recursos transformados, quando entram em operação (ciclo), são pré-selecionados de acordo com a característica construtiva do produto, com intuito de seguir um fluxo específico dentro de uma célula, onde a matéria-prima, pessoas e equipamentos (recursos transformadores) necessários para a transformação de um produto encontram-se disponíveis. [Slack *et. al.*, 1999].

Segundo Corrêa e Corrêa, 2006, este tipo de layout busca aumentar a eficiência do processo, comparado ao arranjo físico funcional. No entanto, deve-se ter cuidado para que este aumento na eficiência não acabe influenciando na flexibilidade do processo. O arranjo físico celular baseia-se nos conceitos de tecnologia de grupo (TG). Com base nesses conceitos, os recursos não similares se agrupam de maneira que seja possível processar um grupo de itens que necessitam de etapas similares de processamento. Desta forma, as principais vantagens esperadas em um arranjo celular são:

- aumento na velocidade e eficiência, pois os recursos existentes nas células estão próximos;
- redução nas distâncias percorridas pelos fluxos dentro das células;
- redução nos tempos de preparação dos equipamentos nas células, já que processam itens de forma e dimensões similares;
- melhoria no controle da produção, pois cada célula é focalizada num grupo de itens;
- manutenção da flexibilidade, pois o mesmo conjunto original de itens continua sendo processado;
- melhoria na qualidade do produto, pois são os funcionários que gerenciam e operam os recursos das células, criando uma sensação de “propriedade” e não apenas de uma etapa produtiva.

Lorini, 1993, ainda complementa que o arranjo celular é o tipo que possui maior tendência de utilização pela engenharia industrial, pois este tipo de arranjo reúne as vantagens dos arranjos em linha e funcionais, pois as máquinas são dispostas próximas umas das outras, seguindo uma sequência mais adequada para o processo, obtendo-se flexibilidade não mais para um só produto, mas para uma família de peças.

A figura 2.5 ilustra a representação de um arranjo físico celular, onde, em uma célula pode-se arranjar máquinas de características diferentes, visando à fabricação de um produto ou uma família de peças.



**Figura. 2.5 - Disposição de um Layout Celular**

Este tipo de arranjo físico é indicado em situações em que se tem uma variedade de peças média e um tamanho de lote de pequeno a médio. Além disso, cumpre registrar que quanto mais estável for a demanda utilizada e quanto maior for o ciclo de vida dos produtos fabricados mais adequada se torna a sua utilização. De acordo com Gonçalves Filho *et. al.*, 2004, a implementação de um arranjo físico celular requer a formação de grupos de máquinas, ou seja, é preciso definir quais as máquinas que pertencerão a cada células e a disposição destas máquinas dentro da células.

Destaca-se ainda que o projeto de um layout celular deve ser realizado de maneira que o objetivo principal durante a etapa de planejamento seja a otimização e a formação de famílias de peças e grupos de máquinas, onde os principais parâmetros de projeto a se considerar são [Lorini,1993]:

- roteiros de fabricação;
- lista de materiais;
- lista de máquinas (capacidade e tempos);
- demanda requerida ao sistema;
- know-how da mão de obra.



## 2.3 Técnicas de Formação de Células

### 2.3.1 Tecnologia de Grupo

Com relação às técnicas de formação de células pode-se dizer que o conceito de Tecnologia de Grupo (TG) é essencial para suas eventuais aplicações, pois segundo Lorini, 1993, a Tecnologia de Grupo sob aspecto de manufatura aplica-se como filosofia de organização, notadamente agrupando as peças e recursos para sua fabricação. Para Black, 1999, estes recursos agrupados, no caso, as células de manufatura devem ter as máquinas arranjadas da melhor maneira para processarem uma família de peças. Groover, 2011, conceitua que a Tecnologia de Grupo consiste em uma filosofia de produção em que se identificam as peças similares, que são agrupadas para tirar vantagens de suas similaridades com relação a produção e projeto, assim estas peças, similares, são agrupadas em famílias específicas as quais possuem características de projeto e produção similares. Singh e Rajamani, 1996, complementam que o conceito de Tecnologia de Grupo surgiu para reduzir configurações de tamanhos de lotes e as distâncias de percurso, ou seja, a tecnologia de grupo tenta manter a flexibilidade de um layout funcional com a alta produtividade de um layout do tipo em linha.

De acordo com a literatura [Lorini,1993; Groover, 2011], existem diversos benefícios que a Tecnologia de Grupo pode trazer para um ambiente fabril:

- controle mais eficiente do projeto do produto;
- ausência ou minimização de estoques intermediários;
- planejamento de processos e cronograma de produção são simplificados;
- melhor acesso e precisão das informações;
- racionalização da programação da produção;
- tempos de preparação são reduzidos, resultando em tempos de processamento mais baixos;
- redução de estoques ao longo do processo;
- redução do *Lead Time* (tempo entre a entrada e saída do material no chão de fábrica);
- promove a padronização das ferramentas, dos sistemas de fixação e das preparações dos equipamentos.

Para a implementação dos princípios da Tecnologia de Grupo, toda uma reorganização do ambiente de manufatura é necessária, onde as etapas que devem ser consideradas neste processo são:

- adequação do “*layout*” físico;
- implantação de um sistema de codificação e classificação;
- racionalização de projetos e processo de fabricação;
- formação de famílias de peças e células de máquinas.

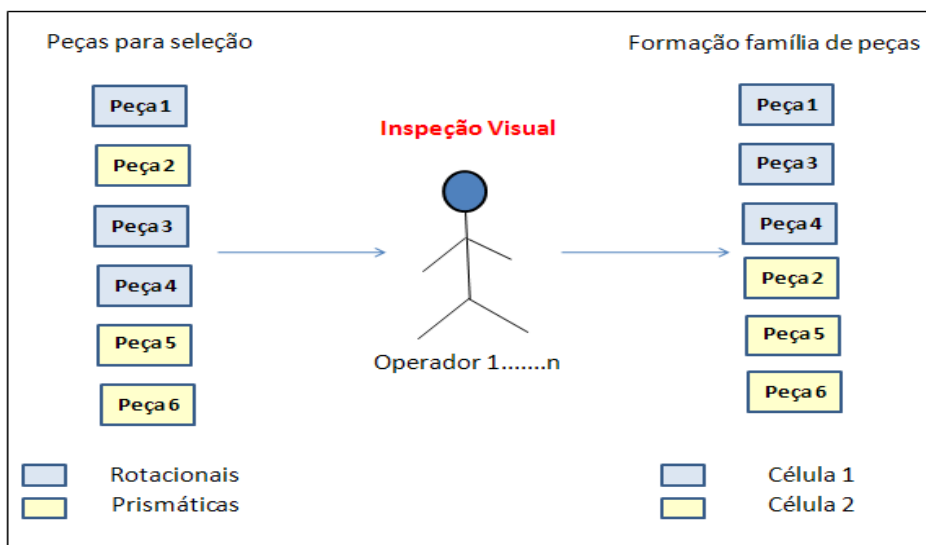
Segundo Groover, 2011, existem três abordagens gerais para resolver o problema de agrupamento de família de peças. Estes métodos requerem um determinado tempo para sua implementação e envolvem a análise dos dados por uma equipe devidamente treinada, sendo:

- inspeção visual;
- classificação e codificação através da análise dos dados de geometria e processo;
- análise do Fluxo de Produção (PFA).

No entanto, para formar um arranjo físico do tipo celular, inicialmente é necessário estabelecer qual critério deve-se utilizar para agrupar itens e formar famílias de produtos. Os critérios mais utilizados são: similaridade, fluxo de processo, classe de produtos, categoria de máquinas ou formas geométricas [Brito, 2010]. Lorini, 1993, acrescenta que não existem regras rígidas ou gerais para determinar que peças serão agrupadas em que famílias, ou seja, cada usuário específico pode definir seus próprios critérios, conforme seus objetivos.

### 2.3.2 Inspeção Visual

O método de inspeção visual consiste em agrupar as peças em famílias, através da análise visual das características das peças. Trata-se de um método simples que não necessita de grandes custos para sua implementação. [Groover, 2011]. Segundo Lorini, 1993, este método é o menos eficiente de ser altamente dependente da experiência humana, além de ser limitado pelo número de peças que fisicamente podem ser manuseadas em tempo aceitável, sendo sua aplicação viável somente no caso de pequenos números de peças.



**Figura. 2.6 - Representação do método de Inspeção Visual**

### 2.3.3 Codificação e classificação de peças

A metodologia de um sistema de classificação consiste na codificação de peças onde atributos geométricos, de processo, ou outros, sejam representados de modo claro e preciso. Isto significa que uma vez as peças estando codificadas, através dos critérios de similaridade estabelecidos, é possível inferir sobre os códigos para determinar os agrupamentos.

Segundo Groover, 2011, o sistema de classificação e codificação (SCC) é de implementação mais demorada e complicada que os métodos de inspeção visual e as técnicas relacionadas à análise do fluxo de produção (PFA). Muitos sistemas foram desenvolvidos, mas nenhum tem sido universalmente adotado. Uma das razões para isso é que um sistema de classificação e codificação deve ser especificamente projetado para uma determinada empresa ou indústria, afinal os produtos de cada empresa são específicos de acordo com a produção realizada. Black, 1999, complementa que trata-se de um método caro e que leva tempo para sua implementação.

De acordo com Lorini, 1993 e Singh e Rajamani, 1996, os códigos que são gerados podem ser estruturados em três formas básicas, que são:

- Estrutura Monocódigo: este tipo de código é conhecido como uma estrutura do tipo árvore, onde cada dígito amplia as informações do seu anterior, isto é, o valor do dígito no código tem seu significado relacionado ao seu termo anterior, não podendo ser analisado de maneira separada.

- Estrutura Policódigo: neste tipo de código cada símbolo é independente dos caracteres anteriores, ou seja, cada símbolo caracteriza um parâmetro. Além disso, este tipo de estrutura de código se caracteriza pelo fato de não haver hierarquia para sua construção. Este tipo de estrutura é considerável adequada para aplicações de classificação e codificação, no entanto existem casos em que é necessário uma elevada quantidade de símbolos.
- Estrutura Combinada: a estrutura que consiste na associação de pequenos monocódigos interligados a policódigos.

Diversos métodos utilizados para a classificação e codificação de peças baseiam-se em estruturas combinadas, destacando-se os seguintes métodos:

- Sistema Optiz – desenvolvido por H. Optiz na Universidade de Aachen, Alemanha;
- MICLASS (*Metal Institute Classification System*) – desenvolvido na Holanda pela TNO (*The Netherlands Organization for Applied Scientific Research*);
- Sistema KK-3 – desenvolvido no Japão pela JSPMI (Sociedade Japonesa para Promoção de Máquinas Industriais);
- Sistema SCC/GRUCON – desenvolvido pelo Grupo de Pesquisas e Treinamento em Comando Numérico e Automação Industrial.
- Sistema C/C – desenvolvido pela John Deere Company;
- MultiClass – desenvolvido pela Organização para Pesquisa Industrial (OIR), USA.

A figura 2.7 ilustra o modelo para representação básica de geração de código para peças definido por Optiz, uma concepção de estrutura combinada.

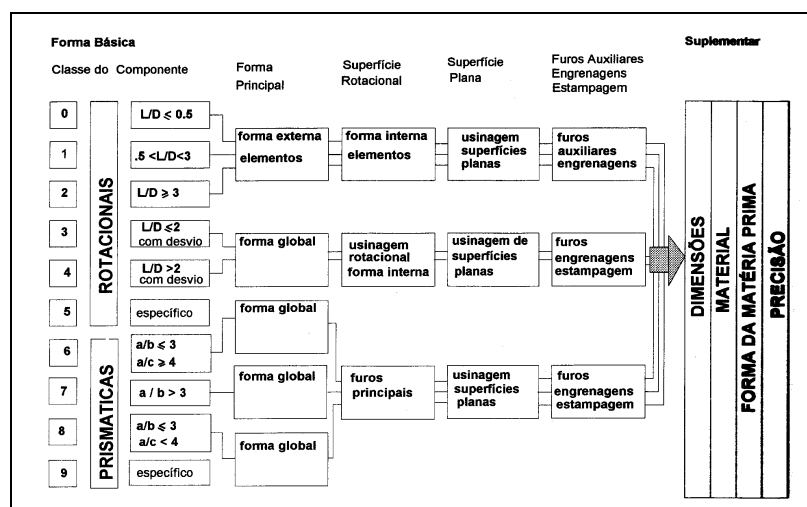


Figura. 2.7. Representação básica do sistema Optiz - adaptado de Groover, 2011

Como se pode observar cada dígito compõe uma característica de uma peça:

- Dígito 1: Representa se a peça é rotacional ou prismática;
- Dígito 2: Indica a forma principal da peça;
- Dígito 3: Representa a superfície rotacional;
- Dígito 4: Representa a superfície plana;
- Dígito 5: Indica se a peça possui furos auxiliares, engrenagens ou estampagens;
- Dígito 6: Relacionado com as dimensões da peça;
- Dígito 7: Relacionado ao material utilizado;
- Dígito 8: Representa a forma inicial da matéria-prima;
- Dígito 9: Indica a incerteza dimensional da peça.

A geração de família de peças por classificação, através de um sistema de código é um método eficaz, utilizado em grandes empresas, mas não é abordada neste trabalho.

#### 2.3.4 Análise do Fluxo de Produção (PFA)

A Análise do Fluxo de Produção consiste em uma técnica que analisa a sequência de operação e percurso das peças através das máquinas e estações de trabalho em um ambiente de chão de fábrica. Assim, as peças são agrupadas de acordo com as coincidências dos fluxos de processo definidos para cada componente, independente do tamanho ou formato geométrico. [Lorini, 1993].

Segundo Black, 1999, a Análise do Fluxo de Produção é uma ferramenta importante na reorganização de sistemas, pois através deste método é possível realizar uma análise preliminar da relação custo/benefício antes de sua implementação, além de permitir a implementação de uma determinada célula sem esperar até que todas as peças na fábrica tenham sido codificadas.

Conforme Singh e Rajamani, 1996, uma das grandes vantagens na utilização do método PFA é a sua simplicidade, além de se obter resultados de uma maneira relativamente rápida.

Para, Groover, 2011, a Análise do Fluxo de Produção utiliza dados de manufatura ao invés de dados de projeto para identificar as famílias de peças. Assim, a desvantagem de utilizar a Análise do Fluxo de Produção é que este método não prevê um mecanismo para racionalização dos roteiros de fabricação, além de não fornecer informações se as rotas utilizadas são boas ou consistentes.

O princípio básico para se definirem famílias de peças e grupos de máquinas com base no fluxo de produção consiste em se tabular as informações na forma de uma matriz de incidência peça-máquina. Esta matriz é representada através da tabulação das informações de fluxo, considerando a relação peça-processo na forma de linhas-colunas da matriz, conforme demonstrado na figura 2.8.

		Peça									
		1	2	3	4	5	6	7	8	9	10
Máquina	1	1			1		1			1	
	2		1	1				1			
	3				1	1	1				1
	4		1		1	1				1	1
	5	1		1			1		1		1

**Figura. 2.8 - Representação de uma Matriz de incidência peça-máquina**

Algoritmos são aplicados sobre a matriz com objetivo de se evidenciar peças com semelhanças de processos, caracterizando-se como formadores de uma "família", enquanto os grupos de máquinas correspondentes formam células.

Outro ponto a se considerar com relação a esta matriz é o critério de representação da incidência, ou seja, existem diversas maneiras de assinalar na matriz, como, símbolos gráficos (x,\*), tipo binário ("0","1") e valores decimais (0.1,0.2, ...,n), que caracteriza o respectivo tempo de processamento em uma determinada peça e máquina.

### 2.3.5 Algoritmos e Técnicas de Agrupamento

Conforme Lorini, 1993, os algoritmos podem ser classificados em diferentes categorias, em função das metodologias em que se baseiam. Os algoritmos convencionalmente são do tipo: programação matemática, técnicas de agrupamento hierárquico, técnicas baseadas em arranjo e técnicas de agrupamento não hierárquico (heurísticos).

#### 2.3.5.1 Programação Matemática

Trata-se de algoritmos relacionados nos mais diversos métodos de programação linear, programação quadrática, p-mediana, etc. Estes algoritmos possuem objetivos muito específicos na análise dos resultados, como de minimizar os custos ou tempos, através de informações de seqüência de operações em uma matriz peça-máquina. [Lorini, 1993].

De acordo com Singh e Rajamani, 1996, os métodos heurísticos podem ser afetados pelos dados de entrada e da matriz inicial, em função de que estes dados de entrada não sejam bons para a análise desejada, salientando que existe uma necessidade de desenvolver modelos matemáticos que podem proporcionar soluções eficientes, visando estabelecer uma base de comparação com os métodos heurísticos.

#### 2.3.5.2 Técnicas de Agrupamento Hierárquico

De acordo com Singh e Rajamani, 1996, agrupamento (*Clustering*) é um nome genérico para uma variedade de métodos matemáticos que podem ser utilizados para quais os objetos de um conjunto que possuem semelhanças. Esta técnica tem sido utilizadas em diversas linhas de pesquisa, em áreas como, medicina, biologia e agrupamentos de máquinas. [Singh e Rajamani, 1996].

As técnicas de agrupamento hierárquico estão relacionadas com as metodologias que consideram a similaridade ou dissimilaridade entre as máquinas em relação a sua utilização nas peças a serem processadas. Como exemplo dessas técnicas de agrupamento hierárquico pode-se citar o SLC (*Single Linkage Clustering*), que segundo Lorini, 1993, é um algoritmo baseado num coeficiente de similaridade entre dois elementos das células de máquinas, que serve de parâmetro para o agrupamento. Desta forma, conhecendo o coeficiente de similaridade entre as máquinas, e estabelecendo um valor limite representativo da similaridade, estarão definidos os agrupamentos dos grupos de máquinas e as respectivas famílias de peças, segundo esses parâmetros.

#### 2.3.5.3 Técnicas baseadas em arranjo

Segundo Brito, 2010 as técnicas baseadas em arranjo visam formar grupos de máquinas e associar peças às famílias pelo rearranjo adequado no posicionamento das linhas e colunas, visando transformar a matriz peça-máquina na forma diagonalizada em blocos, onde os elementos dos blocos agreguem semelhanças de fluxo na matriz peça-máquina.

Existem diversas técnicas baseadas em arranjo utilizadas para a formação de células, como por exemplo, ROC (*Rank Order Clustering*), BEA (*Bond Energy Algorithm*), ROC 2 (*Rank Order Clustering 2*), MODROC (*Modified Rank Order Clustering*), DCA (*Direct Clustering Algorithm*) e CIA (*Cluster Identification Algorithm*), entre outros.

### Método ROC - *Rank Order Clustering*

O método ROC é considerado um dos métodos mais clássicos e importantes, que foi desenvolvido por King, 1980. [Lorini, 1993, Singh e Rajamani, 1996 e Groover, 2011]. O método consiste em estabelecer uma ponderação sobre a matriz binária, onde “1” representa a incidência de uma peça na máquina e “0” representa a inexistência de incidência de uma peça na máquina. De posse destes valores na matriz, o algoritmo ROC rearranja as linhas e colunas da matriz de modo iterativo em “n” alterações de forma que as linhas e colunas fiquem dispostas em ordem decrescente.

A figuras 2.9 ilustra um exemplo de Singh e Rajamani, 1996, onde verifica-se inicialmente o ordenamento das linhas, através da representação binária na ordem decrescente dos valores. Na figura.2.10 é realizado o mesmo procedimento, porém considera-se o ordenamento das colunas da matriz analisada. Assim, alterando-se a posição das respectivas linhas e colunas a matriz resulta na forma diagonalizada, que consiste na formação de células, figura 2.11.

	Valores Binário →	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>		
		Peças						Decimal	
		1	2	3	4	5	6	Equiv.	Ordem
Máquinas	01			1		1		10	5
	02		1	1				24	4
	03	1			1			36	2
	04		1	1		1		26	3
	05	1			1		1	37	1

**Figura. 2.9. Matriz de incidência peça-máquina: Ordenamento das linhas [Brito, 2010]**



↓ Valores Binário		Peças					
		1	2	3	4	5	6
$2^4$	Máquinas 05	1			1		1
$2^3$	03	1			1		
$2^2$	04		1	1		1	
$2^1$	02		1	1			
$2^0$	01			1		1	
Decimal		24	6	7	24	15	16
Equivalente.							
Ordem		1	5	4	2	6	3

**Figura. 2.10. Matriz de incidência peça-máquina: Ordenamento das colunas [Brito, 2010]**

↓ Valores Binário		Peças						Decimal Equiv.    Ordem	
		$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$		
$2^4$	Máquinas 05	1	4	6	3	2	5	56	1
$2^3$	03	1	1					48	2
$2^2$	04				1	1	1	7	3
$2^1$	02				1	1		6	4
$2^0$	01				1		1	5	5
Decimal		24	24	16	7	6	5		
Equivalente.									
Ordem		1	2	3	4	5	6		

**Fig. 2.11. Matriz de incidência peça-máquina: Solução ROC [Brito, 2010]**

### Método BEA - *Bond Energy Analysis*

O método BEA (*Bond Energy Analysis*). visa relacionar objetos a atributos, através de uma avaliação de uma medida de efetividade (ME) para verificar quais os elementos possuem melhor afinidade. Assim, os elementos da matriz são reorientados, alternando-se as linhas e as colunas. Esta metodologia consiste, primeiramente, em rearranjar as colunas, buscando posicionar os elementos em posições em que seja maximizado o valor de ME. Este procedimento é realizado da mesma forma visando o rearranjo das linhas, resultando, após a realização destas etapas a obtenção da matriz na forma diagonalizada.

A medida de efetividade (ME), utilizada neste algoritmo, é definida de maneira que, quanto maior for o número de elementos (densidade) em um arranjo, maior é o valor correspondente da medida de efetividade. Isto quer dizer que a medida de efetividade está relacionada com um elemento e seus respectivos vizinhos na corrente de disposição. Assim, a avaliação de um arranjo de ordem  $A(m,n)$  é definido pela seguinte equação:

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

Com relação a este algoritmo, cumpre registrar que ele pode ser aplicado para qualquer tipo e tamanho de arranjo, no entanto existe a restrição de que os elementos não podem ser negativos.

### **Método ROC 2 (*Rank Order Clustering 2*)**

Segundo King and Nakornchai *apud* Singh e Rajamani, 1996, o algoritmo ROC 2 foi desenvolvido devido às limitações computacionais encontradas na metodologia ROC. Este algoritmo consiste no arranjo das linhas e colunas, através da identificação dos elementos da matriz que possuem valores de entrada "1", sendo que o uso de palavras binárias não é considerada nesta metodologia, mas a idéia de classificação de forma ordenada ainda permanecem com outras limitações.

O algoritmo, que ilustra a metodologia ROC 2, é apresentado a seguir:

Passo 1: Realiza-se o arranjo das linhas. De  $p = P$  (a última coluna), localize as linhas de entrada de 1, a seguir mova as linhas com as respectivas entradas para o topo da lista das linhas, mantendo a ordem anterior de entradas. As linhas são analisadas em ordem decrescente das colunas.

Passo 2: Realiza-se o arranjo das colunas. De  $m = M$  (a última linha), localize as colunas de entrada de 1, a seguir mova as colunas com as respectivas entradas para o topo da lista das colunas, mantendo a ordem anterior de entradas. As colunas são analisadas em ordem decrescente das linhas.

Passo 3: Repete-se os passos 1 e 2 até que nenhuma mudança ocorra.

A matriz da figura 2.12 ilustra a implementação do algoritmo na diagonalização de uma matriz de 6 máquinas e 8 peças [Singh e Rajamani, 1996].

		Peças							
		1	2	3	4	5	6	7	8
Máquinas	1	1	1			1			1
	2			1	1		1	1	
	3	1	1	1		1	1		1
	4				1			1	1
	5			1	1			1	
	6	1	1				1		

**Figura. 2.12 - Matriz de incidência peça-máquina: Método ROC 2**

Inicialmente, analisa-se a última coluna ( $p=8$ ), verificando se cada elemento da matriz contém a entrada "1". Assim, como pode se observar as máquinas 1,3 e 6 são movidas para o topo da lista, enquanto as linhas 2,5 e 4 permanecem na mesma ordem. Realizado este procedimento, o mesmo deve ser repetido para as demais colunas da matriz, considerando que a leitura é realizada da esquerda para a direita. A matriz com as linhas permutadas pode ser observada na figura. 2.13.

		Peças								
		1	2	3	4	5	6	7	8	
Máquinas	3	1	1	1		1	1		1	1
	1	1	1			1			1	2
	6	1	1				1			3
	2			1	1		1	1		4
	5			1	1			1		5
	4				1			1	1	6

Novo número de linhas

**Figura. 2.13 - Matriz de incidência peça-máquina: Ordenamento das linhas**

Por conveniência renumera-se novamente essas linhas no intervalo de 1 a 6 a partir do topo. Assim, há uma nova linha na realidade corresponde a linha antiga 3, e assim por diante.

Por fim, realiza-se o arranjo das colunas, onde o processo é realizado da mesma forma, porém é selecionada a última linha para verificar os elementos da matriz que contêm "1". Na Fig. 2.14 observa-se uma matriz onde as linhas e colunas foram rearranjadas através da metodologia ROC 2.

		Peças								
		1	2	8	5	6	3	4	7	
Máquinas	3	1	1	1	1	1	1			1
	1	1	1	1	1					2
	6	1	1			1				3
	2					1	1	1	1	4
	5						1	1	1	5
	4			1				1	1	6
		1	2	3	4	5	6	7	8	

**Figura. 2.14 - Matriz de incidência peça-máquina: Ordenamento das linhas e colunas**

### Método MODROC (*Modified Rank Order Clustering*)

Conforme Chandrasekaran e Rajagopalan *apud* Singh e Rajamani, 1996, o método ROC tem uma tendência em recolher todos elementos "1" no canto superior esquerdo. Então, ao remover este bloco de colunas da matriz e aplicando o método ROC novamente, o método MODROC recolhe outro conjunto de elementos "1" no canto esquerdo superior. Este processo é realizado até que não haja elementos dispostos na matriz. Logo, este processo irá identificar famílias de peças de maneira mutuamente exclusivas, mas pode conter máquinas que estejam sobrepostas. Em função deste problema, um método de agrupamento hierárquico é aplicado com base em uma medida de associar pares de grupos de máquinas. Este método é encerrado quando os grupos não se intersectam ou quando um único grupo é formado.

### Método DCA (*Direct Clustering Algorithm*)

Chan e Milner *apud* Singh e Rajamani, 1996, propuseram o método DCA, que visa reorganizar as linhas com as células mais à esquerda positivas, isto é, "1" para a parte superior e as colunas com as células positivas na parte mais alta para a esquerda da matriz. Wemmerlov (1983) forneceu uma correção para o algoritmo original para obter resultados consistentes.

O algoritmo, que ilustra a metodologia DCA, é o apresentado:

Passo 1: Inicialmente é feita a soma dos elementos que possuem "1" em todas as linhas e colunas da matriz. A seguir, as colunas são arranjadas em ordem decrescente, ordenando-os em uma sequência a partir do último elemento (mais à direita). Este procedimento também é

realizado para o arranjo das linhas, porém o ordenamento é feito de modo crescente em uma sequência a partir do último elemento (mais inferior).

		Peças						
		1	2	3	4	5	6	
Máquinas	1	1	1		1			3
	2			1		1		2
	3	1			1			2
	4			1			1	2
	5			1			1	2
		2	1	3	2	1	2	
		Soma das colunas						

**Matriz Original**

		Peças						
		3	6	4	1	5	2	
Máquinas	5	1	1		1			2
	4			1		1		2
	3	1			1			2
	2			1			1	2
	1			1			1	3
		3	2	2	2	1	1	
		Soma das colunas						

**Matriz rearranjada**

→

**Figura. 2.15 - Rearranjo das linhas e colunas da matriz - Etapa 1**

Passo 2: Nesta etapa, inicia-se analisando a primeira coluna da matriz, onde as linhas que possuírem algum elemento igual a "1" são movidas para o topo. Assim, este procedimento é realizado para todas as colunas da matriz, logo, se em algum caso houver linhas com o elemento igual a "1" no bloco não é necessário movê-lo. No entanto, podem haver casos em que as linhas não pertencem ao bloco, então estas linhas são utilizadas para formar um novo bloco, que é movido para a parte inferior. Uma vez que os elementos estiverem atribuídos a um bloco, a linha não é movida, assim, não é necessário analisar todas as colunas. Na figura 2.16 observa-se um exemplo de uma matriz rearranjada.

		Peças					
		3	6	4	1	5	2
Máquinas	5	1	1				
	4	1	1				
	3	1				1	
	2			1	1		
	1			1	1		1

**Figura. 2.16 - Rearranjo das linhas e colunas da matriz - Etapa 2**

Por fim, deve-se a mover todas as colunas para a esquerda, iniciando pela primeira linha da matriz, um processo similar ao apresentado na etapa 2. Na figura 2.17, observa-se a matriz diagonalizada através da metodologia DCA.

		Peças					
		3	6	5	4	1	2
Máquinas	5	1	1				
	4	1	1				
	3	1		1			
	2				1	1	
	1				1	1	1

**Figura. 2.17 - Matriz de incidência peça-máquina: Solução DCA**

#### **Método CIA (*Cluster Identification Algorithm*)**

De acordo com Singh e Rajamani, 1996, Iri, 1968, propôs um método mais simples para identificar a diagonalização em blocos perfeitos, caso seja possível em função da característica da matriz utilizada. Este método é baseado na teoria da técnica da máscara (*Masking technique*), que consiste analisar todas as linhas e colunas da matriz, verificando se o elemento da matriz possui entrada igual a "1". Assim, estes elementos são mascarados até que os números de linhas e colunas parem de aumentar, constituindo um bloco. No entanto, se não existirem blocos perfeitos, toda a matriz é mascarada como um grupo.

O algoritmo, que ilustra a metodologia CIA, é apresentado como:

Passo 1. Seleciona-se aleatoriamente uma linha  $m$  da matriz, e assinala-se uma linha horizontal  $H_m$  sobre esta linha.

Passo 2. Para cada entrada igual a "1" no cruzamento com a linha horizontal  $H_m$ , assinala-se uma linha vertical  $V_p$ .

Passo 3. Para cada entrada encontrada em uma linha vertical  $V_p$ , assinala-se uma linha horizontal  $H_m$ .

		Peça						
		1	2	3	4	5	6	
Máquina	1	1	1		1			H1
	2			1		1		
	3	1			1			H3
	4			1			1	
	5			1			1	
		V1		V2				
						V4		

**Fig. 2.18. Representação da matriz com os passos 1 -3: Método CIA**

Passo 4. Repetem-se os passos 2 e 3 até que não hajam elementos cruzados cujas as entradas sejam "1".

Passo 5. Transforma-se a matriz original, removendo as linhas e colunas correspondentes a grupos de máquinas e família de peças identificadas no passo 4. Estas Linhas e colunas não aparecem nas iterações seguintes.

Passo 6. Se não houver elementos deixados na matriz, para-se, caso contrário, considerar-se a matriz transformada, assim retorna-se ao passo 1.

		Peça					
		1	2	4	3	5	6
Máquina	1	1	1	1			
	2	1		1			
	3				1	1	
	4				1		1
	5				1		1

**Fig. 2.19. Matriz de incidência peça-máquina: Solução CIA**

#### 2.3.5.4 Técnicas de Agrupamento Não - Hierárquico (Técnicas Heurísticas)

As técnicas heurísticas consistem em procedimentos não classificatórios ou que utilizam coeficientes de similaridade, mas, na maioria de suas aplicações, é utilizado algum critério estabelecido como gerador de agrupamento, via determinados procedimentos heurísticos. [Lorini,1993]. Os métodos heurísticos estão relacionados a um conjunto de regras que visam à resolução de problemas, onde estas regras estão relacionadas com o bom senso do desenvolvedor ou de sua experiência em casos similares. No entanto, um método heurístico não garante a obtenção de uma solução ótima, mas sim uma solução viável. [Tahara *et. al* ,1997].

Entre os métodos heurísticos aplicados na formação de células de manufatura, pode-se citar algoritmos que utilizam *Logica Fuzzy and Clustering* - FC [Kamal & Burke,1996], Algoritmo evolutivo Híbrido [Trindade & Ochi, 2006]. Dentre os métodos heurísticos e meta-heurísticos podem ser destacados os abordados pelos seguintes autores:

Singh e Rajamani, 1996 comenta que as técnicas de Inteligência Artificial mais conhecidas são: RN- Redes Neurais, SA- *Simulated Annealing*, TS - *Tabu Search*, e os AG- Algoritmos Genéticos.

Arzi *et. al.* 2001, propõem uma abordagem de múltiplos objetivos utilizando os AG para a formação de células, que consiste na análise da eficiência e requisitos de capacidade em um ambiente onde a demanda é irregular, ou seja, a capacidade da célula é influenciada pela demanda e correlação entre os tipos de peças atribuídas às células.

Xambre, *et. al.* 2003, propõem um modelo de programação matemática para o problema de formação de células com várias máquinas idênticas, visando minimizar o fluxo intercelular. Devido à natureza do problema, um algoritmo do tipo *Simulated Annealing* (SA) foi apresentado para a resolução do problema.

Coelho, *et. al.* 2006, propõem a aplicação da tecnologia de grupo, através da análise do fluxo de produção, a fim de obter layouts celulares. No entanto para se formar células de manufatura, o trabalho utiliza os algoritmos genéticos, onde consideram matrizes de incidências binárias ou com tempos de processamentos definidos, bem como tamanho de lotes de peças. Além disso, este trabalho utiliza um índice de agrupamento, que avalia o quão eficiente é o agrupamento realizado.

Defersha, *et. al.* 2006, propõem um modelo de programação matemática para a configuração celular e o dimensionamento de lotes em um ambiente que a fabricação é



dinâmica. O modelo também considera o impacto do tamanho de lote sobre a qualidade do produto, de forma que a solução do modelo matemático é minimizar a produção e qualidade dos custos relacionados. Para resolver este modelo para os fins práticos, um algoritmo genético de programação linear foi desenvolvido, onde o algoritmo busca as variáveis inteiras e para cada solução inteira determina-se os valores correspondentes das variáveis contínuas, que são utilizados para a resolução de um subproblema de programação linear.

Arkat, et. al. 2012, propõem o estudo de três modelos heurísticos para o estudo de formação de células, onde analisa-se os algoritmos do tipo Branch and Bound (B & B) e os Algoritmos Genéticos. Os resultados obtidos demonstram a eficácia do algoritmo Branch and Bound (B & B) na solução de grandes problemas de formação de tamanho de células.

Sayadi, et. al. 2013, propõem um modelo baseado no algoritmo *Firefly*, que consiste em um método meta-heurístico, cujo o objetivo principal do artigo é apresentar o algoritmo *Firefly* para resolver problemas de otimização discreta. Para ilustrar o comportamento do modelo e verificar o desempenho do algoritmo, analisa-se um determinado número de exemplos. Este trabalho apresentou que este método é eficaz, de forma que a meta-heurística deve ser útil para futuras pesquisas.

No presente trabalho abordam-se os algoritmos genéticos e suas características para a implementação de células de manufatura.

## **2.4 Algoritmos Genéticos - Caracterização**

### 2.4.1 Introdução aos Algoritmos Genéticos

Os Algoritmos Genéticos (AG) são fundamentados através do processo de seleção proposto por Charles Darwin em 1859. Os Algoritmos Genéticos foram desenvolvidos por John Holland na década de 1960 e posteriormente aperfeiçoados por seus alunos e colegas da Universidade de Michigan em 1960 e 1970. [Michell,1999]. O Objetivo principal de Holland não era projetar algoritmos para resolver problemas específicos, mas sim estudar formalmente como o fenômeno da adaptação ocorre na natureza e desenvolver maneiras em que os mecanismos de adaptação natural possam ser importados para sistemas de computador. O método de Holland foi introduzido no seu livro *Adaptation in Natural and Artificial System* [Holland, 1975].

Segundo Goldberg, 1989, citado por Borges, 2003, observa-se no trabalho de Holland duas metas principais: abstrair e explicar os sistemas adaptativos naturais, além de projetar *softwares* de sistemas artificiais que possibilitem mecanismos importantes de seleção natural.

De forma geral, pode-se dizer que, por meio de um mecanismo iterativo de busca, os Algoritmos Genéticos (AG) objetivam encontrar as melhores soluções para problemas de otimização. Partindo de uma população inicial, o AG promove em um primeiro momento a combinação dos melhores indivíduos; desta combinação, gera uma nova população que substituirá a população anterior, e assim sucessivamente. Deste processo, resulta que uma nova população é gerada a cada iteração realizada. Desta forma, o AG apresenta as novas e melhores soluções para o problema de otimização que está sendo analisado. O processo estará concluído com a convergência do algoritmo [Fernandes, 2005].

Ávila, 2002, define os AGs como algoritmos de otimização estocásticos que funcionam de forma "aleatória orientada" conforme regras probabilísticas que estão relacionadas com os mecanismos da genética natural, isto é, na etapa inicial de implementação do algoritmo gera-se uma população apta para a solução do problema. Então é realizada a etapa de reprodução entre os indivíduos, gerando permutações de material genético através de cruzamentos, e insere-se material genético novo através de mutações. Tudo isto respeitando a lei da genética natural que diz que os mais aptos têm mais probabilidade de sobreviver. Com isto vai-se melhorando a população inicial, sendo que os mais aptos correspondem aos indivíduos que obtêm um valor maior (maximização) ou menor (minimização) em uma equação que representa o objetivo do problema.

Embora os Algoritmos Genéticos sejam simples de descrever e programar, o seu comportamento pode ser complicado, pois existem muitas questões indefinidas sobre o seu funcionamento e sobre a que tipos de problemas são mais aplicáveis. Diversos trabalhos abordam os fundamentos teóricos dos AGs, entre eles, Holland, 1975; Goldberg 1989a; Rawlins 1991; Whitley, 1993b; Whitley and Vose, 1995. [Michell,1999].

#### 2.4.2 Terminologia dos Algoritmos Genéticos

Rao, 2009, afirma que os Algoritmos Genéticos são baseados nos princípios da genética e seleção natural. Os elementos básicos da genética natural, que são conhecidos como reprodução, cruzamento e mutação, são utilizados no procedimento de pesquisa genética.

Segundo Gomes, 2004, as terminologias biológicas para a implementação dos AGs são definidas com base nos conceitos de cromossomo, de genótipo, de fenótipo e de mutação, entre outros.

O autor entende que o Cromossomo consiste em uma a estrutura nucleoprotéica formada por uma cadeia de DNA. Cada espécie possui um número característico de cromossomos que quando são arranjados aos pares, provenientes de cada um dos pais, os indivíduos são chamados de diploides, e os que não se apresentam desta forma são chamados de haploides. Os Genes são blocos funcionais de DNA que são responsáveis pela codificação de uma proteína em específico (G,T,C,A). Cada gene está localizado em uma posição em particular de um cromossomo. Outra característica importante é que pode-se passar o gene como definidor das características do indivíduo. O Genótipo, consiste em um conjunto de genes de um individuo onde dois organismos da mesma espécie podem ter o mesmo genótipo, mas não a mesma carga genética. O Fenótipo é a representação final do potencial do genótipo de certo indivíduo que lhe somará um conjunto de aptidões únicas. A mutação está relacionado com a cópia do material genético dos pais para os filhos. As taxas de mutação ocorrem em maior frequência em indivíduos não adaptados ao ambiente e em menor frequência em indivíduos em harmonia (adaptados) com o ambiente.

De acordo com Goldberg, 1989, os Algoritmos Genéticos estão relacionados aos fenômenos da biologia e à teoria da evolução. Segundo o autor, esta relação pode ser explicada através de adaptações de alguns termos da biologia e de outros conceitos. As definições propostas por Goldberg, 1989, são apresentadas a seguir de forma resumida:

**Genoma:** representa um conjunto completo de genes de um determinado organismo, ou seja, um genoma pode ter diversos cromossomos.

**Cromossomos:** está relacionado com a estrutura de dados que codifica uma solução para um determinado problema, ou seja, um cromossomo representa um ponto no espaço de busca.

**Gen ou Gene:** caracteriza-se pelo parâmetro codificado no cromossomo, isto é, um caractere do vetor (cromossomo).

**Indivíduo:** trata-se de um simples membro da população, onde o indivíduo é formado pelo cromossomo e sua aptidão. Logo, um indivíduo é um conjunto de valores  $x = [x_1, x_2, \dots, x_n]$  para as variáveis de projeto que definem um ponto no qual pode-se avaliar a função objetivo.

**Geração:** consiste no número de iterações que o Algoritmo Genético executa. As gerações representam a evolução das populações ao longo do tempo, em função da aplicação dos operadores genéticos de Reprodução, Recombinação e Mutação.

**Pais e filhos:** em cada geração, o Algoritmo Genético cria populações. Uma população é chamada de pais quando é utilizada para gerar os indivíduos da próxima geração, que são chamados de filhos.

**Genótipo:** no caso dos Algoritmos Genéticos, representa a informação contida no genoma ou cromossomo.

**Fenótipo:** nos AGs, representa o objeto, estrutura ou organismo construído a partir das informações do genótipo, ou seja, trata-se do cromossomo decodificado.

**Alelo:** no caso dos AGs, representa os valores que o gene pode assumir.

**População:** é o conjunto de cromossomos ou soluções. A população consiste na quantidade de indivíduos constituintes de um passo de procura dos AGs.

**Operações Genéticas:** são operações que os AGs realizam sobre cada um dos cromossomos de uma população.

**Recombinação (Crossover):** forma pela qual, através da reprodução, o material genético do indivíduo é misturado, a fim de produzir filhos com características mistas de ambos os pais.

**Mutação:** é responsável pela introdução e manutenção da diversidade genética nas populações. O operador altera a forma aleatória de seu funcionamento do gene de alguns indivíduos, fornecendo meios para introdução de novos indivíduos na população.

**Seleção:** caracteriza-se pelos indivíduos de uma população que são selecionados conforme seu grau de adaptação ao meio.

**Espaço de Busca ou Região Viável:** é o conjunto de espaço ou região que compreende as soluções possíveis ou viáveis do problema a ser otimizado. Deve ser caracterizado pelas funções de restrições, que definem as soluções de forma viável ao problema a ser resolvido.

**Função Objetivo ou de Avaliação:** é a função que se deseja otimizar. Através dela é possível obter a informação numérica do desempenho de cada cromossomo na população. Também é possível representar as características do problema que o AG necessita para cumprir seus objetivos, sendo expresso na forma  $F = f(x_1, x_2, \dots, x_n)$ , onde  $x_1, x_2, \dots, x_n$  são variáveis que o algoritmo procura determinar para otimizar  $F$ . A função objetivo é a

princípio, calculada para cada conjunto de cromossomos de um indivíduo da população e para toda a população.

**Função de Aptidão (*Fitness Value*):** Esta função mostra o quanto é apto um determinado indivíduo perante os outros indivíduos da população daquela geração. É um valor correspondente à função objetivo, avaliada de tal forma que possa ser utilizada pelo algoritmo genético (geralmente a função de aptidão é aplicada à função objetivo de forma que se tenha sempre valores positivos a fim de que, no processo de seleção, haja uma probabilidade de seleção para cada indivíduo).

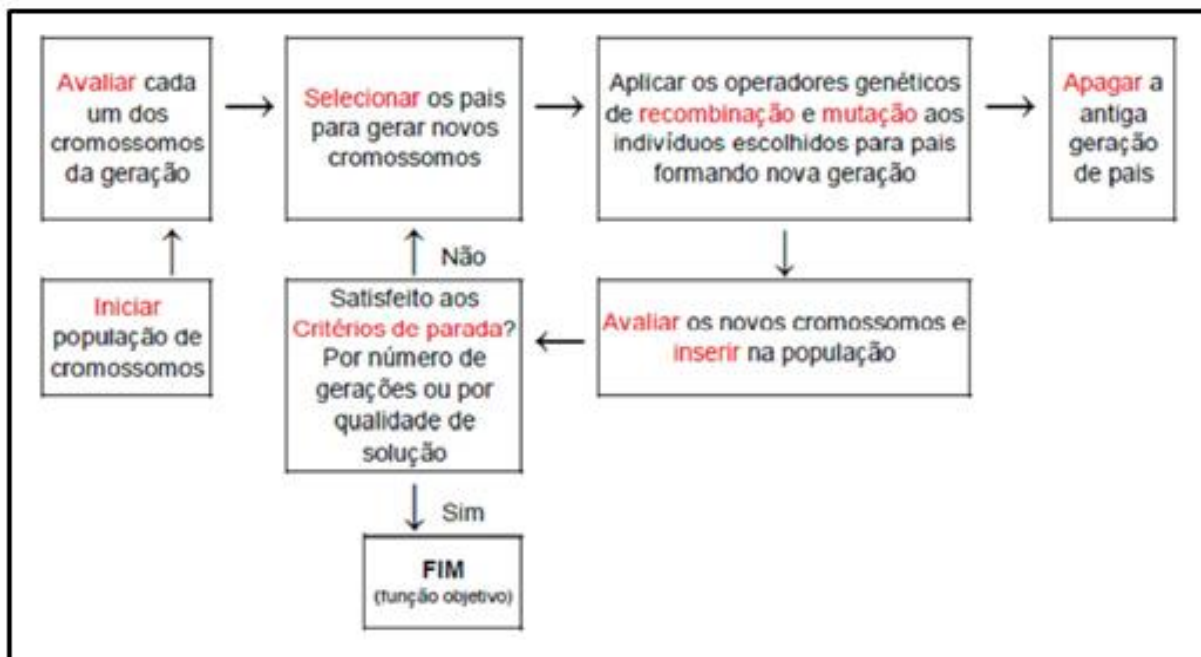
**Ponto Ótimo:** é o ponto caracterizado pelo vetor  $x' = [x_1, x_2, \dots, x_n]^T$ , formado pelas variáveis de projeto que minimizam a função objetivo e satisfazem as restrições do problema.

**Valor Ótimo:** é o valor da função objetivo  $f(x')$  no ponto ótimo.

**Solução Ótima:** é o par, que está relacionado com a solução, formado pelo ponto ótimo e o valor ótimo  $[x', f(x')]$ .

#### 2.4.3 Estrutura dos Algoritmos Genéticos

Os Algoritmos Genéticos operam inicialmente com uma população inicial, onde cada indivíduo (cromossomo) de uma determinada geração é avaliado. Realiza-se a seleção dos pais para gerar novos indivíduos (cromossomos), onde, em seguida são aplicados os operadores genéticos de recombinação (*crossover*) e mutação aos indivíduos definidos como pais para a formação da geração seguinte. Assim, a antiga geração de pais é apagada e os novos cromossomos são avaliados e inseridos na população. Se os critérios de parada são satisfeitos, pelo número de gerações ou por convergência, então obtém-se a função objetivo. No entanto, se estas condições não forem satisfeitas, então seleciona-se novamente os pais para gerar novos cromossomos até que o critério de parada seja satisfeito. A figura 2.20 ilustra o princípio de funcionamento de um Algoritmo Genético.



**Figura. 2.20 - Representação do funcionamento de um Algoritmo Genético (adaptado de Guerra, 2008)**

Inicialmente é necessário especificar as formas de codificar as características de interesse em termos de cromossomos, além de que, no início do processo, deve-se avaliar as características da população inicial. Com relação às condições de critério de parada é preciso avaliar a nova geração em função da geração antiga e dos filhos quanto à sua adaptação. [Gomes, 2004].

No entanto, Borges, 2003, cita que antes de utilizar os Algoritmos Genéticos na resolução de determinados problemas, deve-se analisar a possibilidade de representar as seguintes características:

- o espaço de busca de possíveis soluções para o problema deve ser delimitado a um determinado intervalo de valores;
- uma função objetivo, que indique o quanto uma determinada resposta é boa ou ruim;
- as soluções, possibilitando a codificação na forma computacional.

#### 2.4.4 Características operacionais dos AGs

Segundo Gomes, 2004, algumas características deste método de otimização podem ser assim resumidos:

- trabalham com um conjunto de parâmetros codificados e não diretamente com os parâmetros envolvidos;
- fazem uma busca sobre um conjunto de indivíduos numa população e não sobre um simples indivíduo;
- usam uma função objetivo a ser maximizada ou minimizada e não gradientes como outros métodos como fonte de informação para procura;
- são robustos, podendo ser aplicados em uma grande quantidade de problemas;
- as descontinuidades ou complexidades presentes na superfície da função objetivo a ser otimizada acarretam pouco efeito no desempenho da busca;
- apresentam um desempenho numérico adequado, considerando a grande escala de problemas em que é utilizado;
- sua implementação numérica é geralmente fácil e proporcionam maior flexibilidade no tratamento do problema a ser resolvido.

Segundo Rao, 2009, os Algoritmos Genéticos apresentam diferenças, quando comparados com outros métodos de otimização, como:

- uma população de pontos é utilizada para iniciar o processo, em vez de um ponto único. Se o número de variáveis de projeto é  $n$ , normalmente, o tamanho da população é definido como  $2n$  a  $4n$ . Considerando que vários pontos são usados como candidato para uma determinada solução, os AGs são menos propensos a ficar presos em um ótimo local;
- os Algoritmos Genéticos utilizam somente os valores da função objetivo. Os valores que são derivados não são utilizados no processo de pesquisa;

#### 2.4.5 Operadores Genéticos

Os operadores genéticos realizam sua busca até atingir um resultado satisfatório, ou seja, são estes operadores que modificam a população através das gerações, além de serem necessários para que a população possa se diversificar, preservando as características adaptativas adquiridas nas gerações anteriores. Guerra, 2008, cita que segundo diversos pesquisadores, os AGs sem utilizar os operadores genéticos perderiam o sentido de serem responsáveis por manter a diversidade do indivíduo em uma população.

#### 2.4.5.1 Codificação do cromossomo

De acordo com Argoud *et. al.*, 2008, a codificação está relacionada com a forma que o cromossomo representa uma solução. Um cromossomo  $F = f(x_1, x_2, \dots, x_n)$  é composto por  $n$  elementos, denominados genes, isto significa que cada gene representa uma variável da solução. A codificação pode ser realizada através da representação por números binários (0's e 1's) e números reais. Segundo Ávila, 2002, a representação por números binários foi o primeiro sistema de codificação a ser utilizado, em função de sua analogia direta com a genética natural. Para Guerra, 2008, este tipo de codificação é mais utilizada para problemas que utilizam variáveis discretas, enquanto a codificação por números reais é mais utilizada para problemas que utilizam variáveis contínuas. Deve-se considerar que o tamanho do cromossomo e o tipo de codificação a ser utilizada dependem das características do problema a ser implementado.

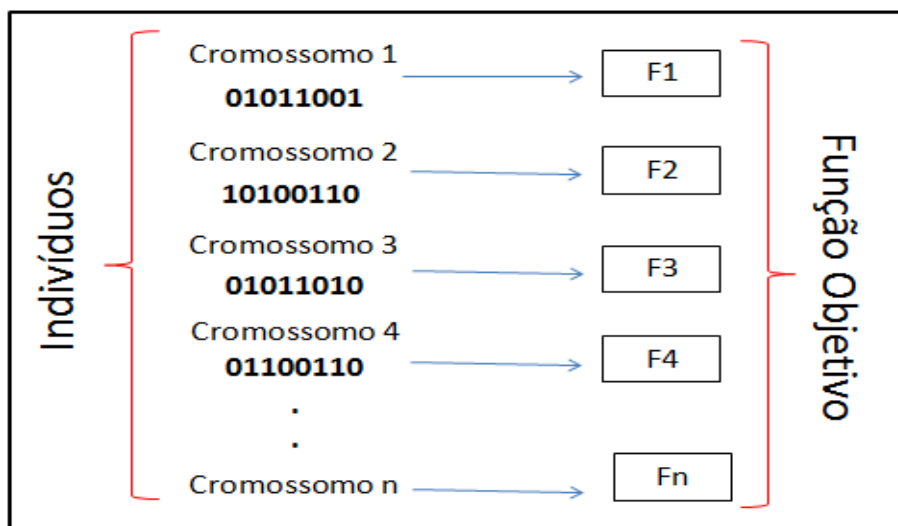
#### 2.4.5.2 Função de Avaliação ou Objetivo

Trata-se da função que avalia o desempenho dos cromossomos em uma população, através de informações numéricas. A função objetivo expressa as características necessárias para a resolução do problema analisado. Sua representação é dada na forma de  $F = f(x_1, x_2, \dots, x_n)$ , onde  $x_1, x_2, \dots, x_n$  são variáveis que o algoritmo procura determinar para otimizar a função objetivo.

Para Souza, 2008, é através da função objetivo que os indivíduos, um a um, recebem as notas em relação à solução encontrada pelo algoritmo. Esta avaliação é que realiza o ordenamento dos indivíduos de forma que o processo de seleção adotado possa escolher os melhores, que conseqüentemente darão origem aos indivíduos da próxima geração.

A resposta obtida da função de aptidão é avaliada em função de um erro, que é calculado através da diferença entre os valores definidos como base e aqueles estimados pelos AGs. Desta forma, o erro, que é calculado a cada iteração, indica o quão boa é a resposta encontrada pelos AGs frente aquelas de referência. Assim, a função objetivo visa analisar a capacidade de cada indivíduo sobreviver devido às variações dos parâmetros envolvidos. A figura 2.21 ilustra os conceitos de aplicação da função objetivo.





**Figura. 2.21- Representação do funcionamento da função objetivo**

#### 2.4.5.3 Seleção de indivíduos

O processo de seleção nos AGs funciona de maneira similar aos fenômenos da natureza, pois os indivíduos que estão altamente adaptados ao seu ambiente possuem, naturalmente, mais oportunidade de se reproduzirem comparados com os indivíduos considerados mais fracos, que consiste na ideia básica da seleção natural apresentada por *Charles Darwin* [Souza,2008].

De acordo com Borges, 2003, após a definição de como os indivíduos (cromossomos) da população estão representados, deve-se escolher o método que irá promover a seleção dos mesmos a fim de gerar outros descendentes. Assim, a proposta deste método é privilegiar os melhores indivíduos na população.

A seleção dos indivíduos de uma população consiste na sobrevivência dos melhores indivíduos, os quais serão selecionados para a reprodução formando uma nova geração. Os principais métodos de seleção são os seguintes:

#### **Seleção pelo Método da Roleta (*Roulett Wheel*)**

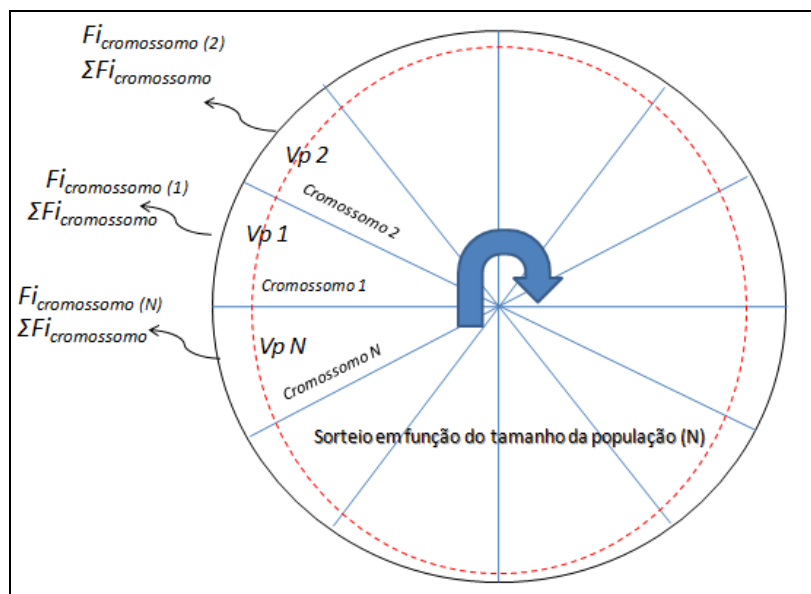
O método consiste na criação de uma roleta composta pelos indivíduos da população, onde cada cromossomo compõe um segmento da roleta proporcional ao valor da função objetivo. Quanto maior for o valor da função objetivo de um indivíduo, maior será a probabilidade deste indivíduo ser selecionado. Este método emprega o princípio da probabilidade de sobrevivência do melhor adaptado, ou seja, a probabilidade  $p_i$  de  $i$ -ésimo

indivíduo da população que vier a ser selecionado para a reprodução é o cálculo proporcional à sua probabilidade de seleção.  $f_i = f(x_i)$ , conforme equação. [Gomes, 2004].

$$P_i = \frac{f_i}{\sum_{i=1}^N f_i} \quad (3.1)$$

Onde  $f_i$  é a função objetivo do elemento  $i$  da população e  $N$  é o tamanho da população

Este método é o mais utilizado e cada indivíduo da população é representado proporcionalmente pelo seu índice de aptidão logo, quanto maior for o índice de aptidão, maior será sua proporção na roleta e, no caso dos indivíduos com menor aptidão, a proporção será relativamente menor. A figura 2.22 ilustra o princípio de funcionamento de uma roleta no caso de uma população de  $N$  indivíduos.



**Figura.2.22 - Seleção via método da Roleta**

### **Seleção pelo Método do Torneio (*Tournament selection*)**

No método de seleção por torneio  $N$  indivíduos (cromossomos) são escolhidos aleatoriamente e o cromossomo que tiver o maior aptidão é selecionado para uma população intermediária. Assim o processo é realizado até preencher a população intermediária, onde geralmente utiliza-se para  $N$  o valor dois. Aumentando o número de indivíduos a serem escolhidos  $N$ , aumenta-se a pressão de seleção, ou seja, cromossomos com aptidão acima da média terão mais chances de serem selecionados [Guerra, 2008]. Neste tipo de método o

indivíduo é escolhido em função de um valor de probabilidade  $k$  que deve ser previamente definida. [Souza, 2008].

Neste método, os melhores indivíduos que possuem maior valor sobrevivem, enquanto os piores morrem. No entanto, Borges, 2003, ainda complementa que os indivíduos que são devolvidos para a população original podem ser selecionados novamente.

De acordo com Gomes, 2004, são vantagens do método de seleção por torneio não acarretar convergência prematura, combater a estagnação, baixo esforço computacional é necessário (ordenamentos) e há uma inspiração biológica no processo que o justifica.

### **Seleção pelo Método Elitista**

O Elitismo, que foi introduzido pela primeira vez por Kenneth De Jong (1975), consiste em um complemento para diversos métodos de seleção, que forçam os AGs a manter alguns dos melhores indivíduos a cada geração. Isto ocorre porque alguns indivíduos podem ser perdidos se não forem selecionadas para a reprodução ou podem ser destruídos pelos operadores de cruzamento ou mutação. O elitismo melhora significativamente o desempenho dos AGs. [Mitchell,1999].

Quando se aplicam os operadores genéticos, existe o risco de se perder um indivíduo de alta aptidão, podendo este não ser criado nas gerações sucessivas. Para evitar que isto ocorra, aplica-se o elitismo, cuja função principal é evitar que os melhores indivíduos desapareçam da população em função da manipulação dos operadores genéticos, isto é os indivíduos que possuem alta aptidão são automaticamente colocados na próxima geração. [Souza,2008].

### **Seleção pelo Método do Ordenamento Linear**

Segundo Guerra, 2008, no método de Ordenamento Linear os indivíduos da população são ordenados em ordem decrescente de aptidão e a aptidão escalonada é dada pela equação:

$$f_i = \min + (\max - \min) \cdot \frac{N - i}{N - 1} \quad (3.2)$$

Sendo

$$1 \leq \max \leq 2 \quad \text{e} \quad \min + \max = 3 \quad (3.3)$$

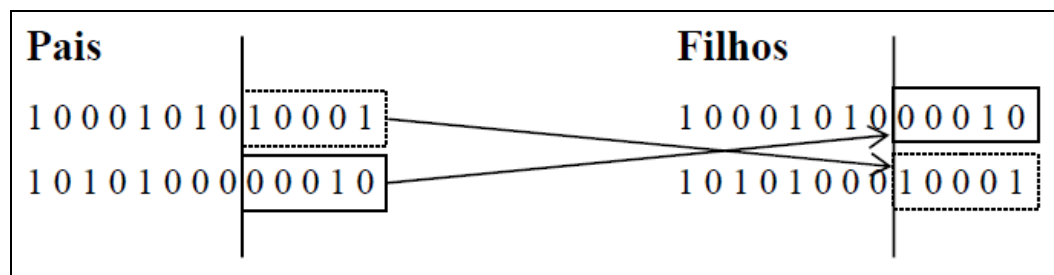
onde  $i$  é o índice do cromossomo na população em ordem crescente;  $N$  é o número de indivíduos na população;  $max$  é a pressão de seleção e  $fi$  é o número de filhos esperados.

#### 2.4.5.4 Operador de Cruzamento (*Crossover*)

Os operadores de cruzamento estão relacionados com os mecanismos de reprodução natural, onde existe a troca de partes do material genético de dois indivíduos (Cromossomos), denominados pais, combinando as informações de maneira que exista uma probabilidade razoável dos novos indivíduos produzidos serem melhores que seus pais. [Souza,2008]. Este método consiste em misturar os materiais genéticos para gerar os filhos. Assim, pode-se realizar a recombinação de todos os indivíduos aos pares para criar os filhos através da recombinação em um ponto ou mais de um ponto.[Silva,2006]

#### ***Crossover* de 1 Ponto**

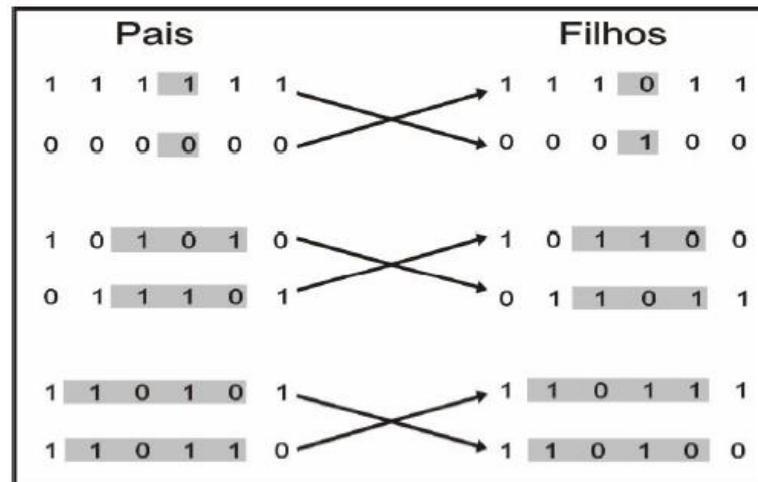
Segundo Borges, 2003, este tipo de *Crossover* consiste em selecionar um ponto de corte de maneira aleatória entre os genes do indivíduo (Cromossomo). Logo, a partir deste ponto, as informações genéticas dos pais são trocadas de forma que as informações anteriores ao ponto escolhido em um dos pais são ligadas às informações posteriores a este ponto no outro pai. Um exemplo de aplicação de *Crossover* foi demonstrado por Silva, 2006 (Figura 2.23).



**Figura. 2.23 - *Crossover* de um ponto na representação binária**

#### ***Crossover* de 2 Pontos**

Segundo Guerra, 2008, o *Crossover* de 2 pontos segue a mesma ideia que o *Crossover* de 1 ponto, agora são gerados dois pontos aleatórios dentro do limite do cromossomo, conforme figura. 2.24.

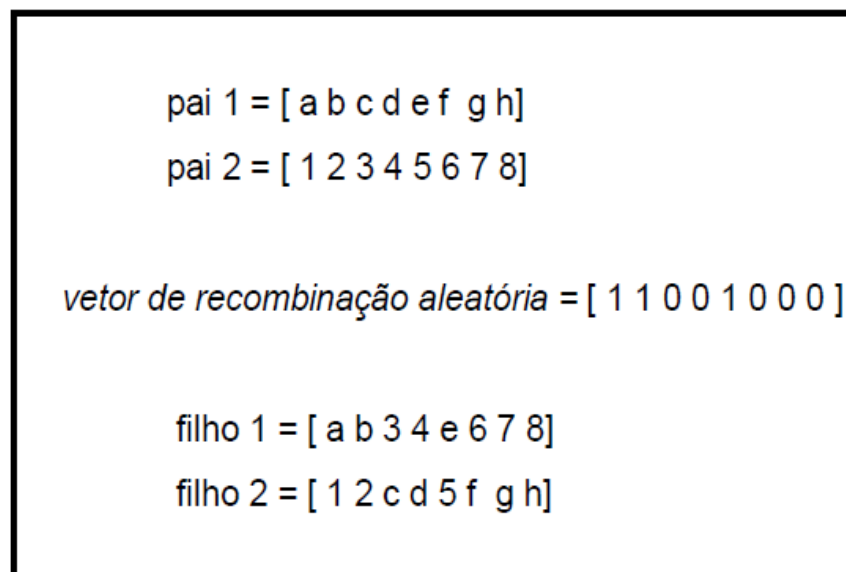


**Figura. 2.24 - Crossover de dois pontos na representação binária (Adaptado de Guerra, 2008)**

### **Crossover Uniforme**

Neste tipo de *Crossover* gera-se uma cadeia binária aleatória que auxiliará a troca do material genético, ou seja, se a cadeia de *bits* é composta pelo *bit* 1, a sua posição correspondente terá seu *bit* alterado no cromossomo dos filhos [Guerra,2008].

Este método não utiliza pontos de cruzamento, mas determina através de um parâmetro global a probabilidade de cada variável ser trocada entre os pais. Além disso, este tipo de *Crossover* se difere dos demais métodos, pois é gerado apenas um filho de cada par de cromossomos, como pode-se observar na figura.2.25.

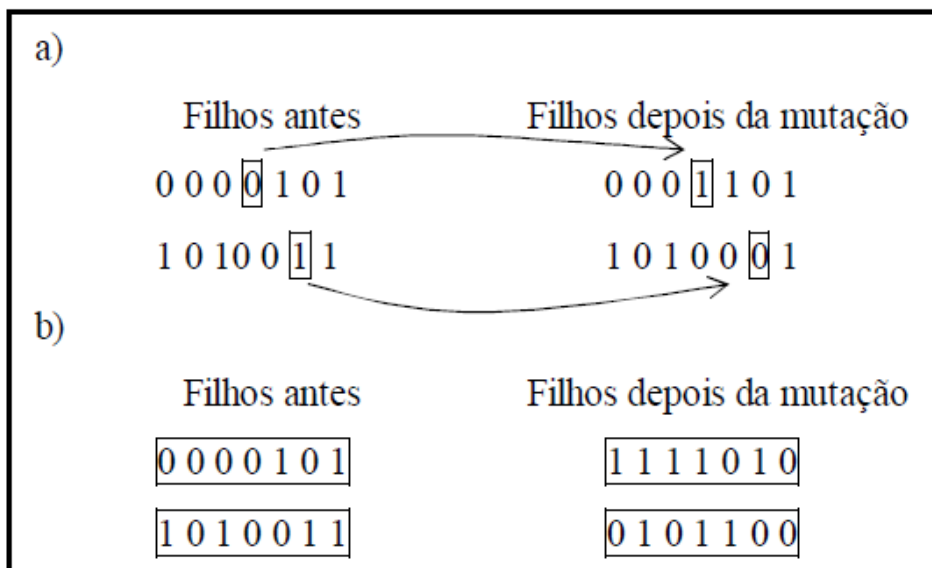


**Fig. 2.25. Crossover Uniforme apresentado por Guerra, 2008.**

#### 2.4.5.5 Mutação

O operador de mutação é aplicado após a operação de cruzamento. Este operador é aplicado com uma determinada probabilidade, em cada *bit* dos dois filhos. Desta forma, o operador de mutação inverte o valor dos genes (*bits*), no caso de 0 para 1 e de 1 para 0. Uma das características da mutação é que ela melhora a diversidade dos cromossomos na população, no entanto, destrói parte da informação contida.

O operador de mutação é responsável pela introdução e manutenção da diversidade genética nas populações. O operador altera de forma aleatória o valor de seu funcionamento de gene de alguns indivíduos, fornecendo assim meios para a introdução de novos indivíduos na população, assegurando que em quaisquer gerações se tenha populações com uma certa diversidade. A mutação pode ser de duas formas: através da substituição *bit a bit* do cromossomo ou pela inversão dos *bits* de todos elementos do cromossomo. Na figura. 2.26 pode-se observar um exemplo de aplicação apresentado por Guerra, 2008.



**Fig. 2.26. Mutação: a) bit a bit b) inversão dos bits.**

### 3 IMPLEMENTAÇÃO DO ALGORITMO NA FORMAÇÃO DE CÉLULAS

#### 3.1 Introdução

Este capítulo apresenta o funcionamento e os recursos disponíveis nos Algoritmos Genéticos que são implementados no ambiente *Matlab*. O *Matlab* é um programa de uma linguagem computacional de alto nível que possui uma boa *performance*, além de ser composto por aplicações específicas chamadas de *toolboxes*, que consistem em compilações de funções pré-existentes mais elementares. O ambiente de programação contém rotinas específicas para resolver problemas de otimização através dos AG.

Mesmo assim é necessário estabelecer algumas modificações nas sub-rotinas do *toolbox* para que o algoritmo genético seja aplicado especificamente no problema de formação de células.

A implementação deste algoritmo no *toolbox* do *matlab* pode ser explicada conforme o fluxograma:

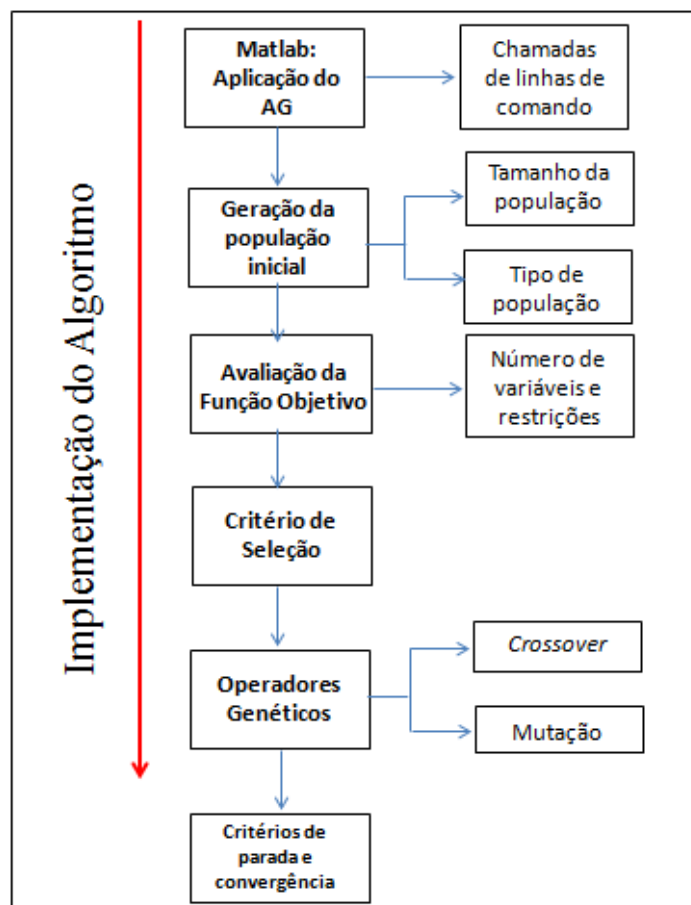


Figura. 3.1 Fluxograma de implementação do Algoritmo.

### 3.2 Matlab: Aplicação dos Algoritmos Genéticos

No ambiente *Matlab*, os AG podem ser implementados de diversas maneiras, onde as formas mais convencionais de aplicações são através de linhas de comando específicas, conforme figura.3.2, ou interface gráficas, como a dos AG em *Matlab*, de acordo com a figura. 3.3. No presente trabalho, optou-se pela utilização de linhas de comando, pois este trabalho objetiva gerar uma interface gráfica específica para iteração com o usuário. Desse modo, a utilização de linhas de comando facilita esta aplicação.

```

%Definição dos parâmetros heurísticos do Algoritmo Genético
options = gaoptimset('PlotFcns',{@gaplotbestf,@gaplotbestindiv,@gaplotexpectation,@gaplotstopping});
options = gaoptimset(options,'PopulationType', 'bitstring');           % Utilizado vetor em bits
options = gaoptimset(options,'PopulationSize',NPOP);                 % População composta por 400 cromossomos (indivíduos)
options = gaoptimset(options,'EliteCount', round(NELIT*NPOP));      % 5% da população já é encaminhado para nova geração
options = gaoptimset(options,'CrossoverFraction', NCROS);          % Taxa de Crossover
options = gaoptimset(options,'Generations', NGER);                 %Numero máximo de gerações
options = gaoptimset(options,'CreationFcn', @gacreationuniform);    %'uniform ' é o comando que gera populações randomicamente
options = gaoptimset(options,'FitnessScalingFcn',@fitscalingrank); %Fitness scaling é proporcional
options = gaoptimset(options,'SelectionFcn', @selectionroulette);  %Método utiizado é do da roleta
options = gaoptimset(options,'CrossoverFcn', @crossoversinglepoint);% Operação de crossover é Single Point
options = gaoptimset(options,'MutationFcn', {@mutationuniform []});% Mutação é Uniforme - Como inserir a taxa
options = gaoptimset(options,'Display', 'off');
%Chamada ao toolbox GA por linha de comando
%X=zeros(nvars,1);
[Xotimo,fotimo]=ga(@ (X) funcao(X,NM,T,NP,ND,NC,MI,nbits,nvars,1),nvars,[],[],[],[],[],[],[],options);

```

Figura. 3.2 - Representação dos AG por linhas de comando em linguagem *Matlab*.

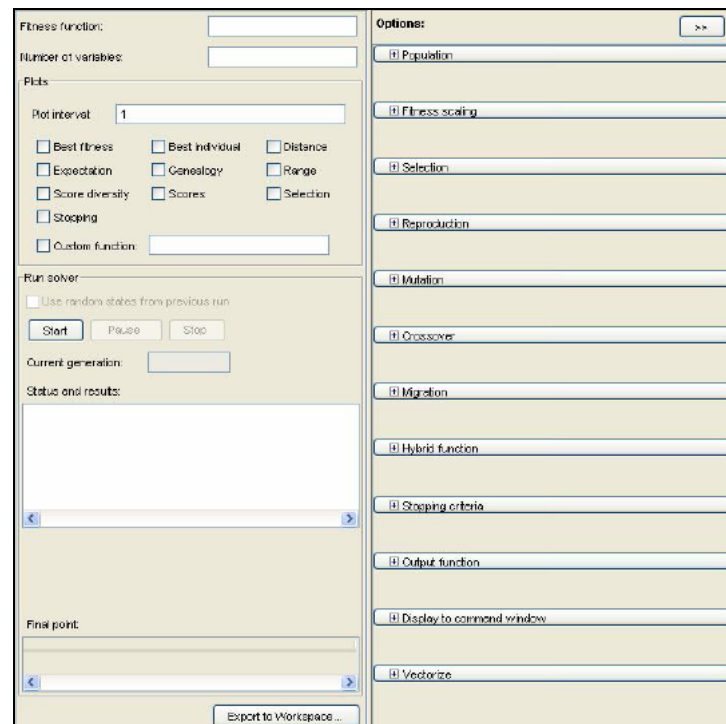


Figura. 3.3 - Interface principal de aplicação dos AG.



### 3.3 Geração da população inicial na formação de células

Inicialmente o algoritmo gera uma população aleatória, cujo tamanho é especificado pelo usuário antes da inicialização do algoritmo, podendo variar este valor de acordo com a diversidade de indivíduos necessária para a implementação desejada.

#### 3.3.1 Tipo de população

Para a formação da população inicial, define-se qual a representação ideal para o cromossomo considerando a aplicação desejada, a função objetivo e os operadores de *crossover* e mutação utilizados. No *matlab* é possível representar a população da seguinte maneira:

- Representação em bits: Também conhecido como *bit string* este comando faz com que todos os elementos do cromossomo (genes) sejam representados por 0's e 1's.

$$\begin{array}{l} \text{Cromossomo} \\ \text{em bits} \end{array} = \left( 01001101101001001 \right)$$

**Figura. 3.4 - Representação de um cromossomo em bits.**

- Representação em números reais: Também conhecido como *double vector* este comando faz com que os elementos dos cromossomos (genes) sejam representados por números reais.

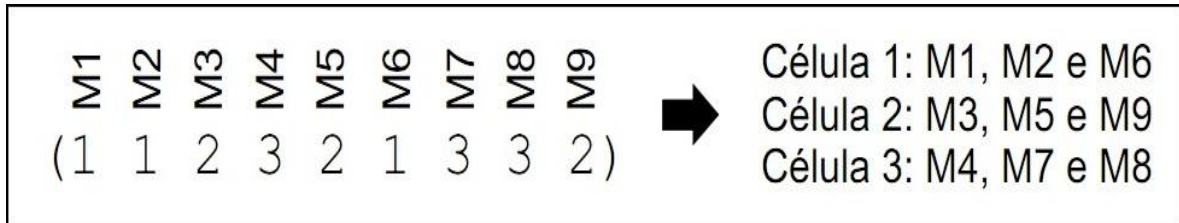
$$\begin{array}{l} \text{Cromossomo} \\ \text{Valores reais} \end{array} = \left( 1 \ 2.5 \ 3.1 \ 2.3 \ 2.7 \ 4.1 \ 4.5 \right)$$

**Figura. 3.5 - Representação de um cromossomo em números reais.**

- Representação Customizada: Também conhecido como *custom type* este comando gera o cromossomo através de uma função específica, determinada para uma aplicação a ser analisada.

Neste trabalho codificou-se o cromossomo de acordo com as representações de Vernugopal e Naredran, 1992 e Gonçalves Filho *et. al.*, 2004, onde cada elemento (gene) está relacionado com a respectiva célula que a máquina pertence, ou seja, cada posição do

cromossomo representa a máquina analisada, enquanto cada caractere representa a célula. Na figura. 3.6 pode-se observar um exemplo de Gonçalves Filho *et. al.*, 2004, que ilustra a codificação utilizada neste trabalho para a formação de três células de manufatura.



**Figura. 3.6** Representação de um cromossomo para formação de células.

Considerando-se que o *toolbox* do *Matlab* não opera com cromossomos representados por número inteiros, para utilizar o *Matlab*, criou-se um algoritmo, implementado como sub-rotina no *toolbox* do *Matlab*, visando realizar a conversão do valor binário do cromossomo para valor inteiro, com intuito de visualizar cada cromossomo na forma desejada (em valores inteiros) na implementação do algoritmo, conforme a figura 3.7.

```

k=0;
for i=1:nbits:nvars
    k=k+1;
    soma=0.0;
    for j=1:nbits
        soma=soma+X(i+j-1)*2^(nbits-j);
    end
    if soma + 1 > NC
        Z(k)=NC;
    else
        Z(k)=soma+1;
    end
end

```

**Figura. 3.7** Sub-rotina de conversão de valores binários para inteiros.

Este algoritmo consiste em analisar todos os elementos de um cromossomo, que é vetor gerado pelo programa. Para que esta transformação ocorra através de forma variável, ou

seja, podendo variar o tamanho da população inicial, torna-se necessário atribuir duas variáveis, conforme as equações a seguir (3.1 e 3.2):

$$nbits = \text{ceil}(\log(NC)/\log(2)) \quad (3.1)$$

Onde: *nbits* define o número de bits como sendo o adequado para gerar os códigos dos número de células;

NC: consiste no número de células em que deseja-se formar.

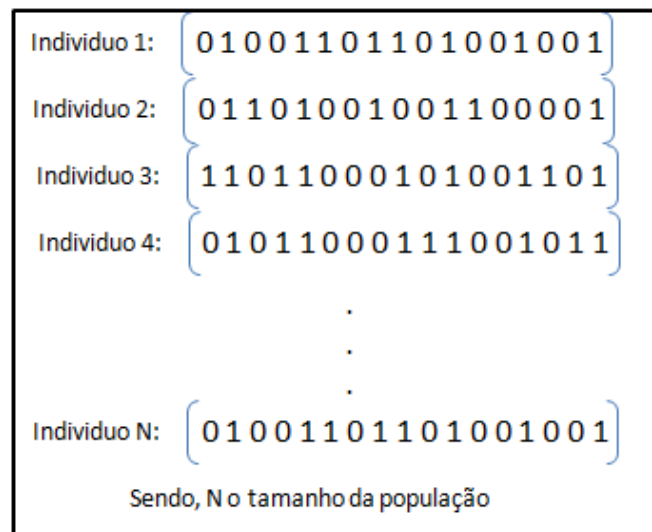
$$nvars = NM * nbits \quad (3.2)$$

Sendo *nvars* a variável que define o número de variáveis binárias que o AG irá trabalhar.

NM: consiste no número de máquinas analisadas no problema.

### 3.3.2 Tamanho da população

O tamanho da população está relacionado com o número de indivíduos (cromossomos) que interagem em cada geração do algoritmo. A figura 3.8 ilustra um exemplo de uma população composta por N indivíduos.



**Figura. 3.8 Representação de um população composta por N indivíduos.**

Neste trabalho considera-se para as análises realizadas valores de população elevados, pois desta maneira aumenta-se a probabilidade de encontrar um indivíduo ideal mas também aumenta-se o custo computacional. por um indivíduo ideal para a formação de células. Os valores utilizados para a população inicial foram obtidos através de testes realizados, verificando o melhor indivíduo, bem como o tempo de simulação do algoritmo.

Em relação à função de criação (*Creation Function*), utilizou-se, para todas as simulações, a opção uniforme (*uniform*), onde os indivíduos são gerados de maneira uniforme de acordo com os limites iniciais colocados.

### 3.4 Função Objetivo (*fitness*)

Entre os estudos de otimização para a formação de células, pode-se observar que a função objetivo é um dos itens mais abordados, a qual busca maximizar ou minimizar um problema relacionado a um objetivo específico.

Segundo Vernugopal e Naredran, 1992, existem vários objetivos importantes associados com o problema de agrupamento de máquinas, mas um primeiro passo, importante na formação de células, é trabalhar com um número gerenciável de objetivos que sejam distintos e mensuráveis.

Desta forma, o presente trabalho aborda os seguintes objetivos:

- F1 - minimização do volume de movimentos entre as células: reduz a movimentação de matéria-prima além de simplificar significativamente o controle de chão de fábrica reduzindo o tempo de fluxo médio e WIP (*Work in process*).
- F2 - minimização da variação da carga total dentro da célula: auxilia de maneira que a variação entre a matéria prima no interior de cada célula seja suave, levando à minimização de WIP (*Work in process*) dentro de cada célula.

De acordo com [Vernugopal e Naredran, 1992; Gonçalves Filho *et. al.*, 2004], o equacionamento que melhor descreve estes objetivos (F1 e F2) pode ser explicado da seguinte maneira:

A função objetivo, que representa cada cromossomo da população, consiste na soma das funções F1 e F2, além de que nesta função consideram-se constantes de proporcionalidade, conhecidas como C1 e C2.

$$F_{\text{indiv\u00edduo}} = C_1 \cdot F_1 + C_2 \cdot F_2 \quad (3.3)$$

#### 3.4.1 C\u00e1lculo da Fun\u00e7\u00e3o F1

A fun\u00e7\u00e3o F1 consiste na equa\u00e7\u00e3o 3.4:

$$F_1 = \sum_{j=1}^n N_j \left[ \sum_{l=1}^k y_{jl} - 1 \right] \quad (3.4)$$

Onde:

$N_j$ : definido por um vetor que relaciona uma determinada demanda (em peças), cujo número de elementos é igual ao número de peças  $j$  analisadas.

$y_{jl}$ : estabelece a relação se a máquina  $i$  possui vínculo com a célula  $l$ , isto é, se  $y_{il}$  é igual a 1 significa que a máquina  $x$  processa uma determinada peça e faz parte da célula  $y$ , conforme a seguinte expressão:

$$y_{jl} = 1 \quad \text{se} \quad \sum_{i=1}^m e_{ji} \cdot x_{il} > 0, \text{ caso contrário, } y_{jl} \text{ é igual a } 0$$

Sendo:

$[e_{ji}]$ : caracteriza-se uma matriz de ordem  $n$  (peça) x  $m$  (máquina), que avalia o processamento peça-máquina, onde  $e_{ji} = 1$  se  $t_{ij} > 0$ , caso contrário  $e_{ji} = 0$ . Cumpre registrar que  $t_{ij}$  (tempo de processamento peça-máquina) representa se a peça  $i$  é processada na máquina  $j$ .

$[x_{il}]$ : consiste em uma matriz binária de ordem  $m$  (máquina) x  $k$  (peça), que avalia a qual célula a máquina pertence, onde  $x_{il} = 1$  se a máquina pertence a célula  $l$ , caso contrário  $x_{il}$  é igual a 0. Para a formulação desta matriz implementa-se no algoritmo uma lógica que realiza a leitura de cada cromossomo e verifica a qual célula ( $l$ ) a máquina ( $i$ ) pertence.

Isto significa para determinar a função F1 uma máquina( $i$ ) deve possuir vínculo com uma célula ( $l$ ), além de que a máquina analisada deve processar uma peça de acordo com a matriz de incidência fornecida inicialmente no problema. A figura 3.9 ilustra o princípio de funcionamento para obter-se a matriz  $x_{il}$  e  $y_{jl}$ , mostrando o princípio de obtenção de F1.

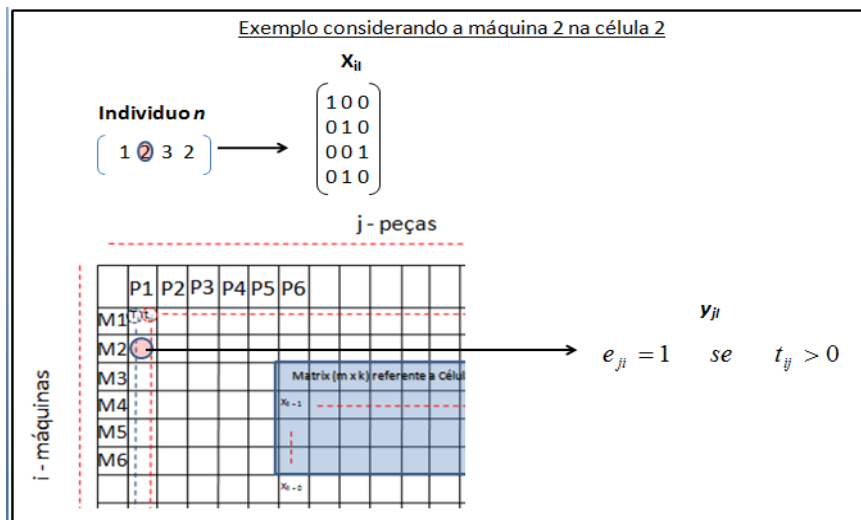


Figura. 3.9 Princípio de funcionamento da função F1.

### 3.4.2 Cálculo da Função F2

A função F2 é expressa pela seguinte equação:

$$F_2 = \sum_{i=1}^m \sum_{l=1}^k x_{il} \sum_{j=1}^n (w_{ij} - m_{lj})^2 \quad (3.5)$$

Onde:

$[x_{il}]$ : consiste em uma matriz binária de ordem  $m$  (máquina) x  $k$  (célula), onde  $x_{il}=1$  se a máquina pertence a célula  $l$ , caso contrário  $x_{il}$  é igual a 0. Para a formulação dessa matriz implementa-se no algoritmo uma lógica que realiza a leitura de cada cromossomo e verifica a qual célula ( $l$ ) a máquina ( $i$ ) pertence.

$[w_{ij}]$ : é definido de acordo com a equação:

$$w_{ij} = \frac{(t_{ij} \cdot N_j)}{T_i} \quad (3.6)$$

A equação 3.7 representa a razão entre o tempo necessário para produzir as peças necessárias (com base na demanda) e o tempo disponível de máquina para o processamento destes lotes.

Onde:

$t_{ij}$ : consiste no tempo necessário para o processamento da peça  $i$  na máquina  $j$ , representado através da matriz de incidência peça-máquina analisada.

$N_j$ : definido por um vetor que relaciona uma determinada demanda (em peças), cujo número de elementos é igual ao número de peças  $j$  analisadas.

$T_i$ : está relacionado ao tempo de máquina disponível para o processamento de um determinado número de peças  $i$  em um dado período de tempo, onde o número de elementos deste vetor é igual ao número de máquinas  $j$  analisadas.

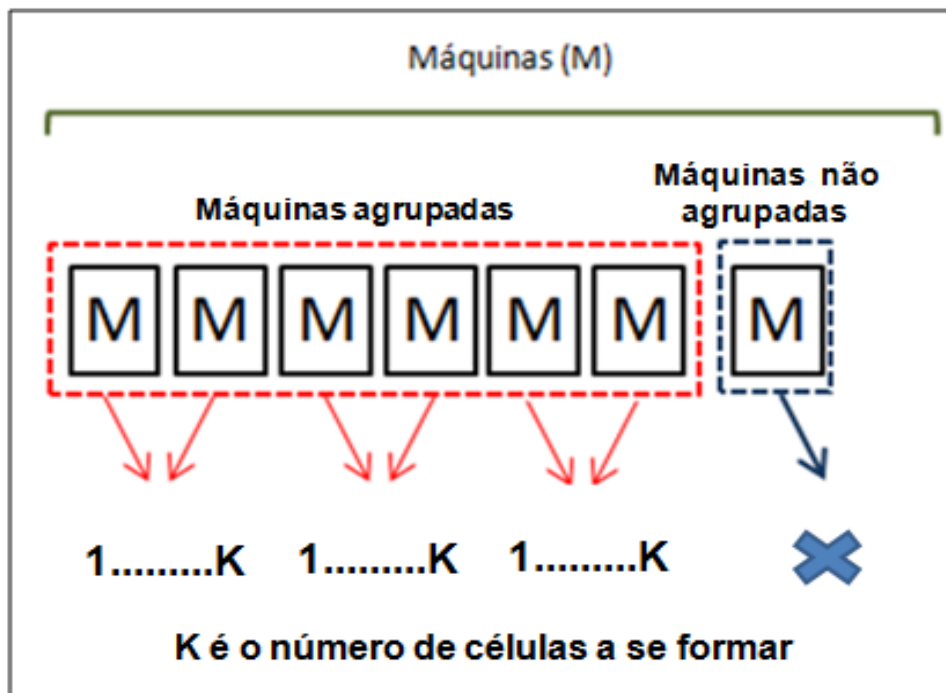
$[m_{lj}]$  é uma matriz de ordem  $k$  (máquina) x (peça)  $n$ , que expressa o carregamento da célula, onde cada elemento é calculado pela seguinte expressão:

$$m_{lj} = \frac{\sum_{i=1}^m x_{il} w_{ij}}{\sum_{i=1}^m x_{il}} \quad (3.7)$$

A função F2 representa que o carregamento de uma célula está relacionado com a demanda existente de uma determinada família de peças, o tempo disponível de um grupo de máquinas para a execução de um processo, o tempo de ciclo para o processamento de peças e a matriz que relaciona as células que a máquina  $i$  pertence.

### 3.4.3 Restrições na formação da população inicial e função objetivo

Conforme comentado no item 3.1.2, a população é gerada de maneira aleatória, e o tamanho da população é definido de acordo com o problema analisado. No entanto, para o caso de formação de células, existem restrições na geração da população que devem ser satisfeitas, ou seja, cada indivíduo que é gerado deve ter a garantia de que não hajam máquinas iguais na mesma célula, além de que cada célula formada seja composta por pelo menos duas máquinas. A figura 3.10 ilustra os requisitos principais para a formação da população inicial, considerando as máquinas disponíveis ( $M$ ) e o número de células a serem formadas ( $K$ ).



**Figura. 3.10** Requisitos para formação da população inicial.

Assim, para implementar estas restrições no *toolbox* do *Matlab*, torna-se necessário acrescentar um algoritmo nas sub-rotinas existentes, capaz de penalizar os indivíduos que não

satisfazem estas condições. Isto significa que a cada indivíduo gerado (cromossomo) é realizada uma leitura dos elementos, verificando a presença de máquinas repetidas ou a existência de uma máquina por célula. Desta forma, quando é identificado um indivíduo com alguma destas características, o algoritmo o penaliza, multiplicando o valor da função objetivo deste elemento por um valor muito elevado, com o intuito de descartar este elemento nas gerações futuras. Na figura 3.11 ilustra-se o algoritmo utilizado para satisfazer os requisitos para formação da população inicial.

```

celula=zeros (NC,1) ;
for i=1:NM
    celula(Z(i))= celula(Z(i))+ 1;
end
penalizacao=0.0;
for j=1:NC
    if celula(j)<2
        penalizacao=penalizacao+10000;
    end
end
F1;
F1=F1+penalizacao;

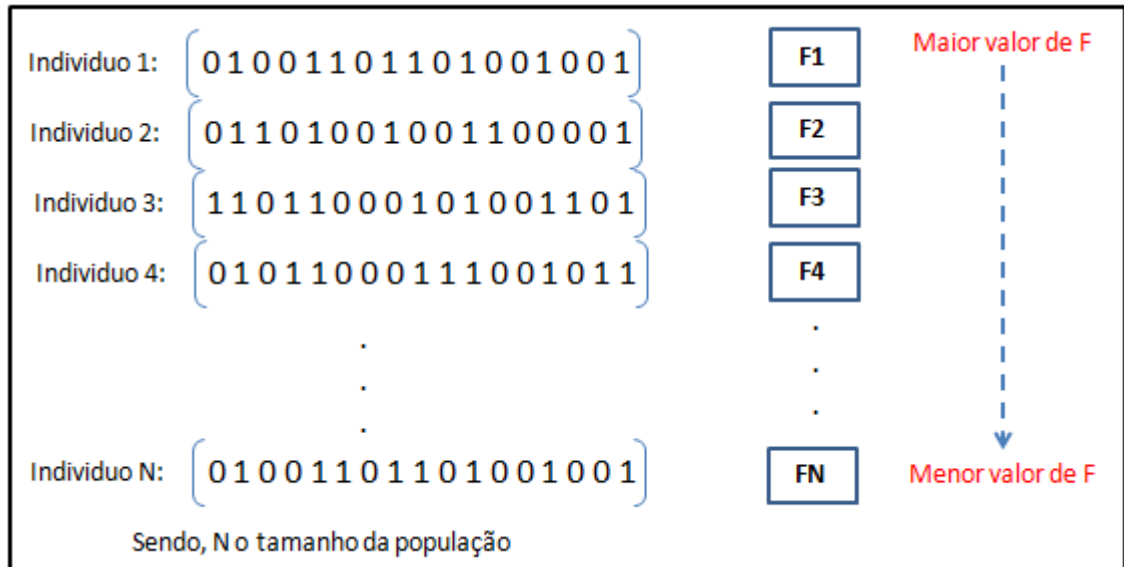
```

**Figura. 3.11 - Algoritmo de restrição da população inicial.**

### 3.5 Escalonamento da função objetivo e Critérios de Seleção

Entre as ferramentas disponíveis no *toolbox* do *Matlab*, é o critério de escalonamento da função objetivo (*scaling fitness*), que consiste no ordenamento dos indivíduos de acordo com o valor da função objetivo obtida. O critério utilizado neste trabalho é conhecido como ordenamento linear (*Rank*), onde os indivíduos são arranjados em ordem decrescente em função dos valores da função objetivo. A figura 3.12 ilustra o princípio de funcionamento desta ferramenta.





**Figura. 3.12 - Princípio de funcionamento da ferramenta por escalonamento**

Após a realização do escalonamento da função objetivo, realiza-se o processo de seleção, onde foi utilizado o método da roleta (*Roulette Wheel*), que é um método muito utilizado em problemas de AG.

### 3.6 Operadores Genéticos

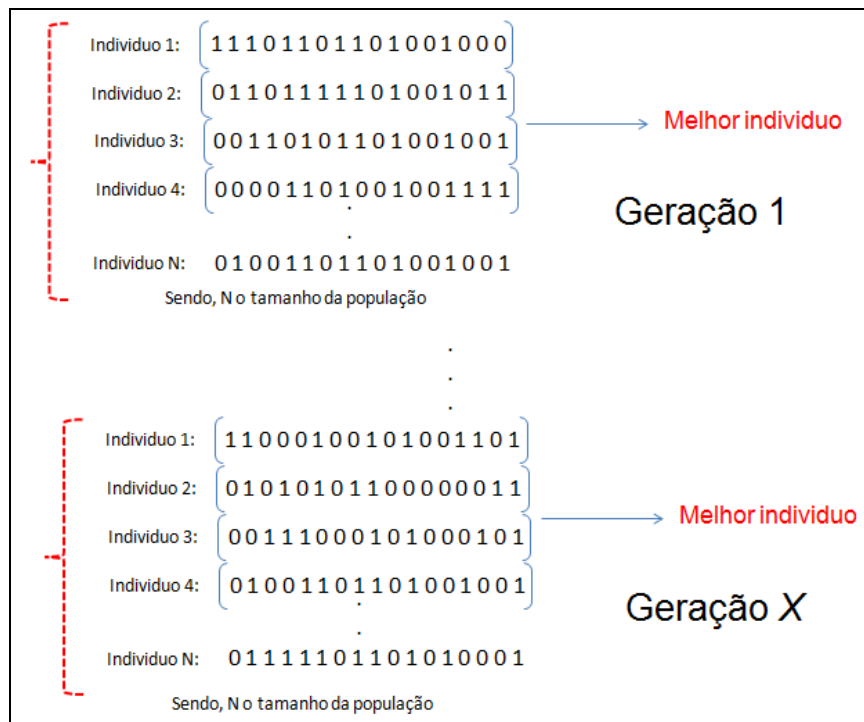
Os operadores genéticos são utilizados quando a seleção dos indivíduos mais aptos para reprodução já foi realizada, logo, o algoritmo gera em sequência uma nova população para a próxima geração, onde os elementos que foram selecionados aplicam os operadores genéticos de Recombinação (*Crossover*) e Mutação da seguinte maneira:

- recombinação (*Crossover*): utilizado para esta análise, o *Crossover* do tipo *single point*, para o qual o valor da taxa de *Crossover* foi modificada;
- mutação: Considerou-se somente a mutação do tipo Uniforme (*Uniform*), modificando somente o valor da taxa de mutação.

### 3.7 Critérios de parada e convergência

Um algoritmo tem como uma de suas principais características o critério de parada, ou seja, o algoritmo finaliza seu processamento quando um determinado critério estabelecido inicialmente é satisfeito. O *toolbox* do *matlab* utiliza a combinação de cinco critérios que podem ser definidos pelo usuário, que são: 1) Valor Limite de Gerações; 2) Tempo Limite; 3)

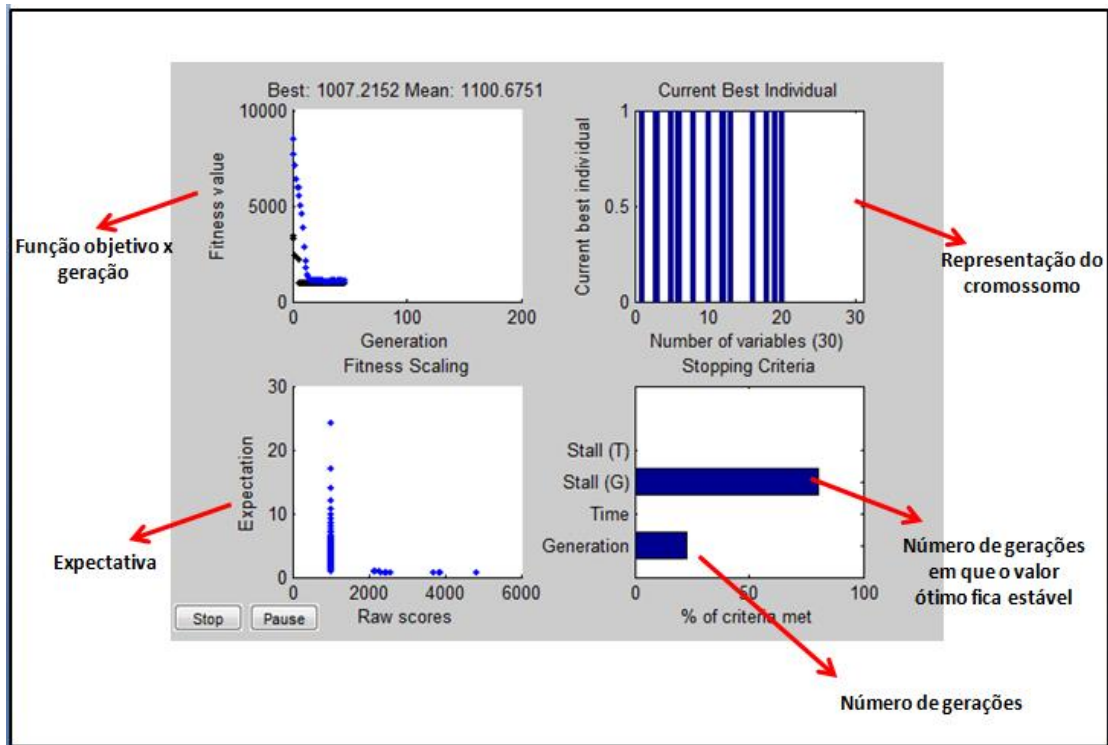
Valor Limite da Função de Aptidão; 4) Número de gerações em que o valor ótimo permanece estável (*stall generations*) e 5) Limite de tempo para a estabilização do valor ótimo entre as gerações (*stall time limit*). Neste trabalho, em cada análise realizada, considerou-se o valor limite de gerações (*generations*) variável, cujo o valor utilizado é de  $X$  gerações de acordo com o problema abordado, enquanto os demais critérios foram utilizados na configuração *Default*. Na figura 3.13 observa-se o critério de parada, devido ao valor limite das gerações.



**Figura. 3.13 - Princípio de funcionamento do AG quando ao critério de parada por gerações**

### 3.8 Monitoramento e análise dos resultados

Outra característica importante na implementação de um algoritmo é monitorar os valores que interessam em relação ao mesmo, visando controlar a forma em que o algoritmo se desenvolve durante as iterações. Neste trabalho, considerou-se a visualização das seguintes características, que são fornecidas no *toolbox* do *Matlab*: 1) melhor função objetivo em uma determinada população (*Best Fitness*); 2) melhor indivíduo naquela população (*Best Individual*); 3) valores relacionados ao critérios de parada (*Stopping*). A figura 3.14 ilustra a representação gráfica dos valores que devem ser monitorados pelo AG



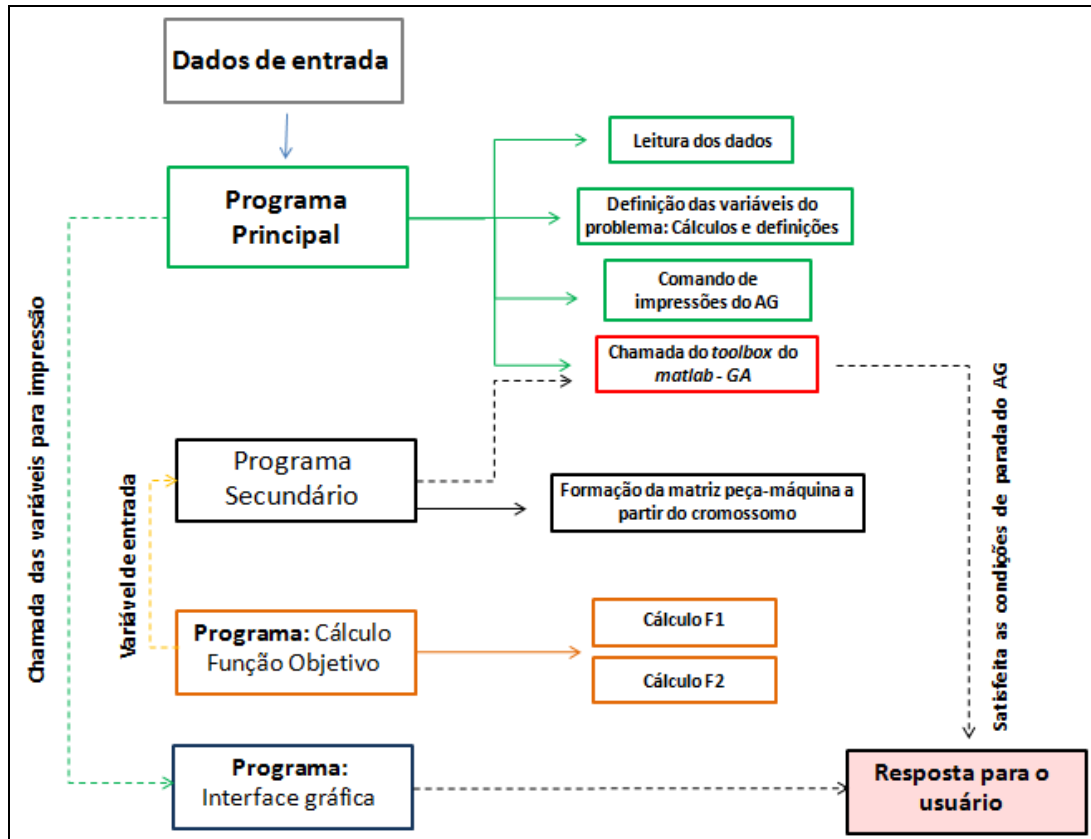
**Figura. 3.14 - Valores monitorados pelo AG.**

### 3.9 Caracterização do Algoritmo proposto aplicado na formação de células

O Algoritmo proposto neste trabalho visa fornecer para o usuário uma interação entre os AG e o processo de formação de células de manufatura. Para que esta interação ocorra, torna-se necessário definir alguns conceitos básicos, como: as características funcionais do algoritmo e o funcionamento da interface gráfica.

#### 3.9.1 Características funcionais do algoritmo

O funcionamento do algoritmo proposto pode ser explicado de forma similar ao funcionamento básico do AG na ferramenta *toolbox* do *Matlab*. No entanto, foram necessárias modificações em algumas sub-rotinas, para as quais foram geradas linhas de comando específicas \*.m em determinados procedimentos realizados pelo algoritmo. A figura 3.15 explica o funcionamento do algoritmo utilizado.



**Figura. 3.15 - Fluxograma de funcionamento do algoritmo proposto.**

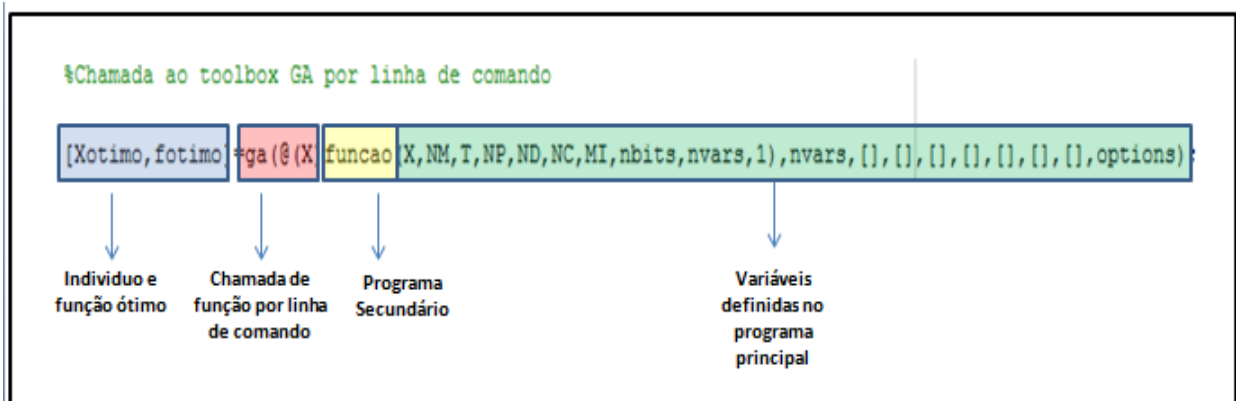
Como pode-se observar no fluxograma, o programa inicia com o fornecimento das variáveis necessárias para a análise dos AG, através de um arquivo \*.dat. Estas variáveis consistem nos seguintes itens:

- número de máquinas;
- número de peças;
- matriz de incidência peça-máquina a ser otimizada;
- tempo disponível de máquina para o processamento de uma família de peças;
- número de células em que se deseja formar;
- tamanho da população (elementos) utilizada em cada geração;
- taxa de *crossover*;
- taxa de mutação;
- número de gerações.

De posse destes valores, o algoritmo realiza a leitura destas variáveis, através do programa principal, utilizando o comando *fscanf*. Outra característica deste programa

principal é que ele define as variáveis do problema, ou seja, são geradas as variáveis que atendem todos os demais subprogramas gerados. Porém, a principal função do aplicativo é a chamada do *toolbox* do *Matlab* para algoritmo genéticos e a impressão dos resultados otimizados após o término do algoritmo. Este procedimento é realizado com linhas de comando específicas, onde são inseridos parâmetros para a simulação do algoritmo. Os parâmetros que são considerados fixos são: tipo de população, número de indivíduos encaminhados para a próxima geração (*elite count*) e o método de seleção utilizado, enquanto os parâmetros definidos pelo usuário são: tamanho da população, taxa de mutação, taxa de *crossover* e número máximo de gerações.

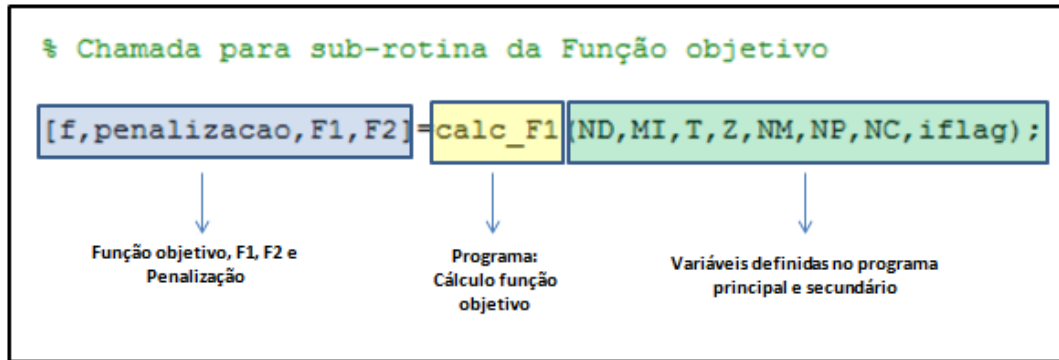
Com os parâmetros estabelecidos em relação as linhas de comando dos AG, o programa principal realiza a chamada para a inicialização do algoritmo, conforme o seguinte comando:



**Figura. 3.16 - Representação da chamada do *toolbox* por linha de comando.**

A ativação do *toolbox* GA é realizada através de uma sub-rotina (programa secundário), que utiliza as variáveis definidas como dados de entrada, enquanto os valores do indivíduo ótimo e função ótima são fornecidos como dados de resposta.

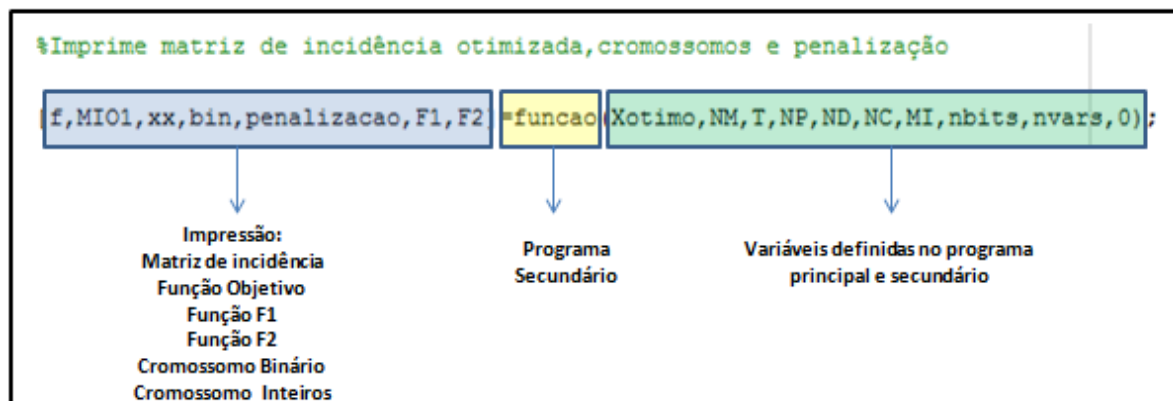
Este programa secundário, ou sub-rotina, inicialmente realiza a conversão do cromossomo (indivíduo) na forma binária para números inteiros, conforme descrito no item 3.3.1. Outro ponto a se considerar deste programa é que ele realiza a chamada das sub-rotinas referentes à função objetivo, isto é, uma linha de comando deste programa chama outra sub-rotina, que realiza o cálculo da função objetivo, conforme ilustra a figura 3.17.



**Figura. 3.17 - Representação da chamada do cálculo da função objetivo**

A sub-rotina secundária (programa secundário) utiliza comandos específicos para realizar impressões de informações como: o cromossomo na forma binária, o cromossomo na representação de células, o valor de penalização caso o indivíduo seja penalizado e a matriz de incidência peça-maquina.

Desta forma, de posse dos valores da função objetivo, com as variáveis e os parâmetros do algoritmo e a chamada para o *toolbox*, o programa busca uma sub-rotina do programa secundário, visando às impressões geradas pelo programa principal.



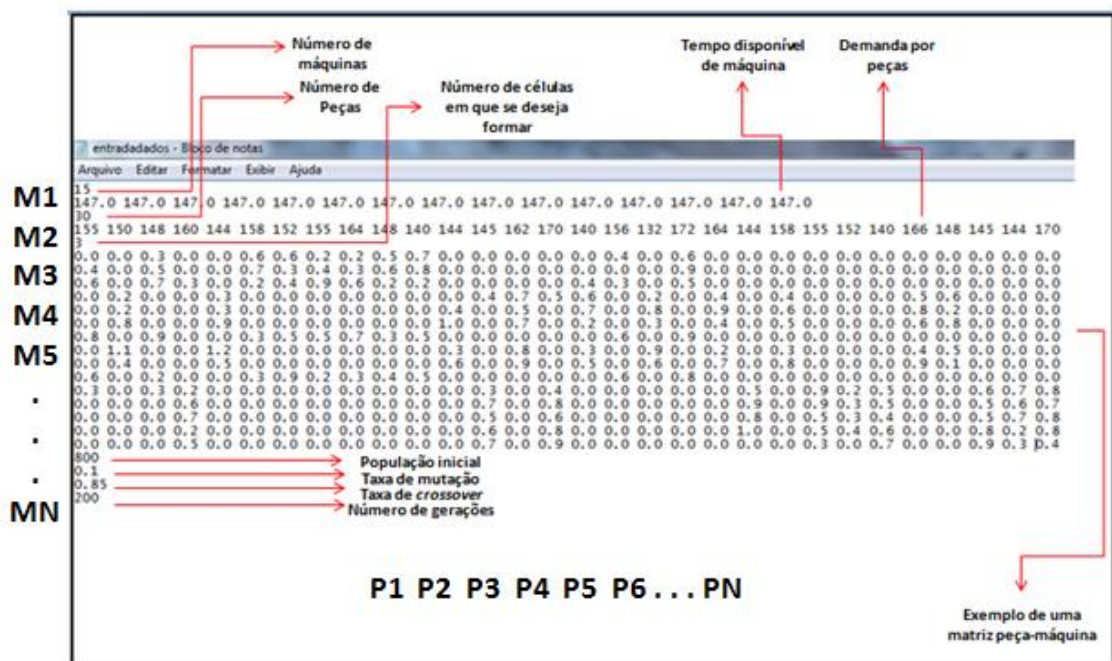
**Figura. 3.18 - Representação da chamada da sub-rotina para impressão.**

Para que o algoritmo forneça os resultados para usuário, gerou-se um programa (sub-rotina) que realiza as impressões dos elementos de saída citados na figura 3.18. Este procedimento é realizado de maneira que todas as variáveis geradas como resposta pelo algoritmo principal sejam enviadas para esta sub-rotina, formatando a impressão dos dados, bem como a interface gráfica, que é apresentada no item seguinte.

### 3.9.2 Funcionamento da interface gráfica

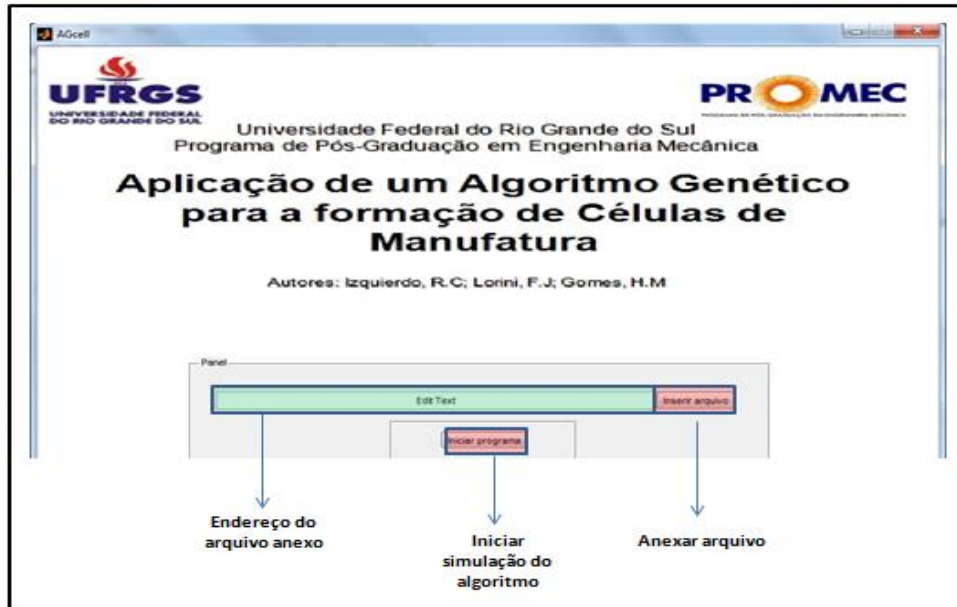
Conforme já enfatizado, este trabalho visa fornecer uma interface funcional para que se possa interagir com os AG, com intuito de formar células de manufatura. Desta forma, criou-se um arquivo com extensão ".exe", cujo o objetivo é importar um arquivo de dados na extensão \*.dat para permitir a verificação das melhores soluções de acordo com o problema analisado.

Para trabalhar com esta interface, inicialmente deve-se preencher um arquivo de extensão \*.dat, onde são inseridas informações, como os parâmetros dos AG e os requisitos básicos para a formação de células. Na figura 3.19, observam-se os elementos, que correspondem a variáveis do problema e podem ser editados no sistema.



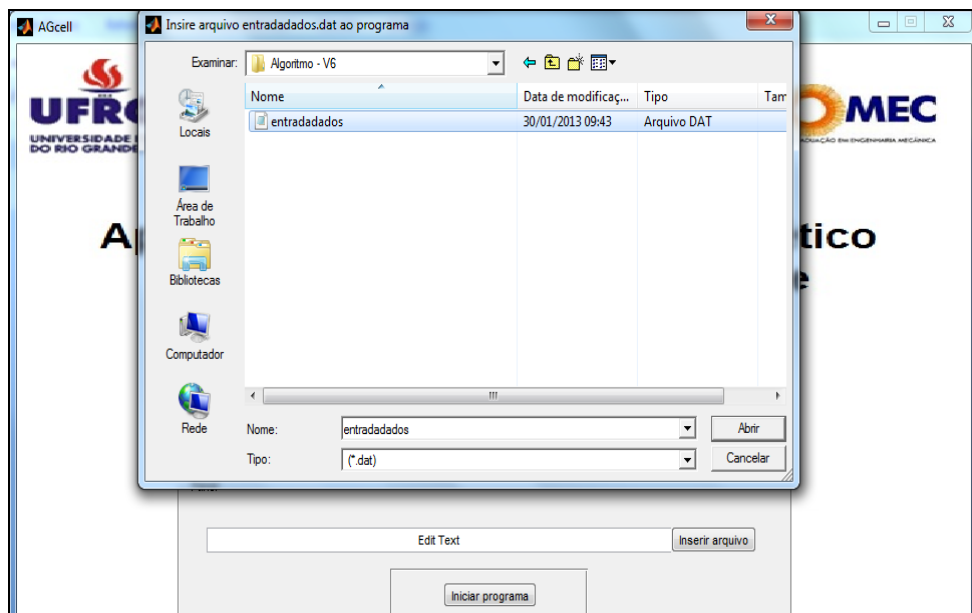
**Figura. 3.19 - Interface do usuário para definição das variáveis**

Após a definição das variáveis que se deseja utilizar para a resolução do problema, no próximo passo deve-se inserir este arquivo com extensão ".dat" no programa executável. A interface principal deste aplicativo é composta por três elementos básicos: uma opção para anexar o arquivo, outra que indica o endereço de onde se encontra o arquivo anexado e uma opção para inicialização do algoritmo. A figura 3.20 ilustra o princípio de funcionamento da interface principal do aplicativo proposto.



**Figura. 3.20 - Interface principal do aplicativo proposto**

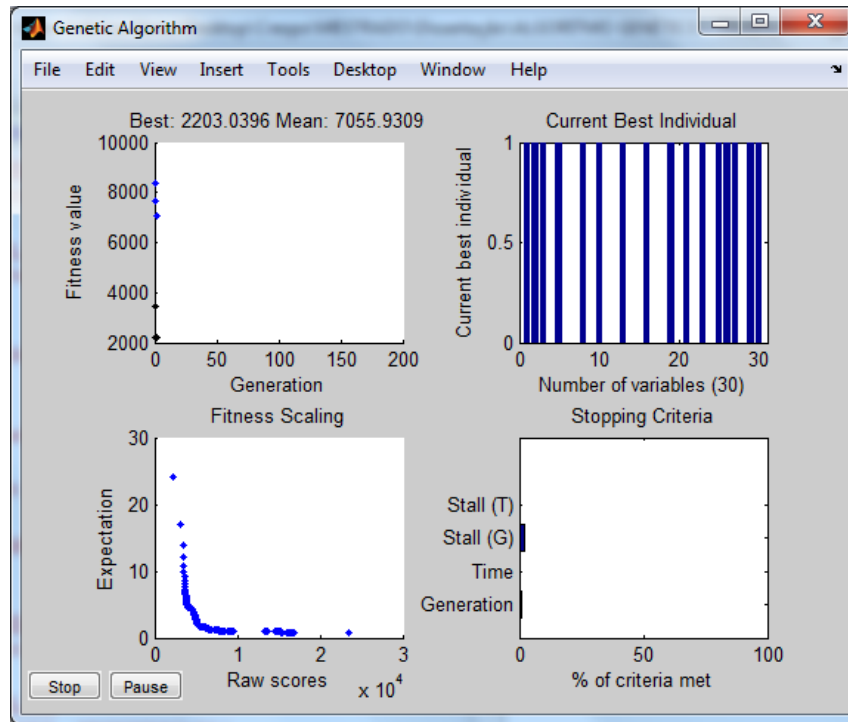
Com a interface principal inicializada, deve-se utilizar a opção **inserir arquivo**, onde uma nova interface permite a busca do arquivo desejado, conforme ilustrado na figura. 3.21.



**Figura. 3.21 - Inserindo o arquivo de dados do usuário**

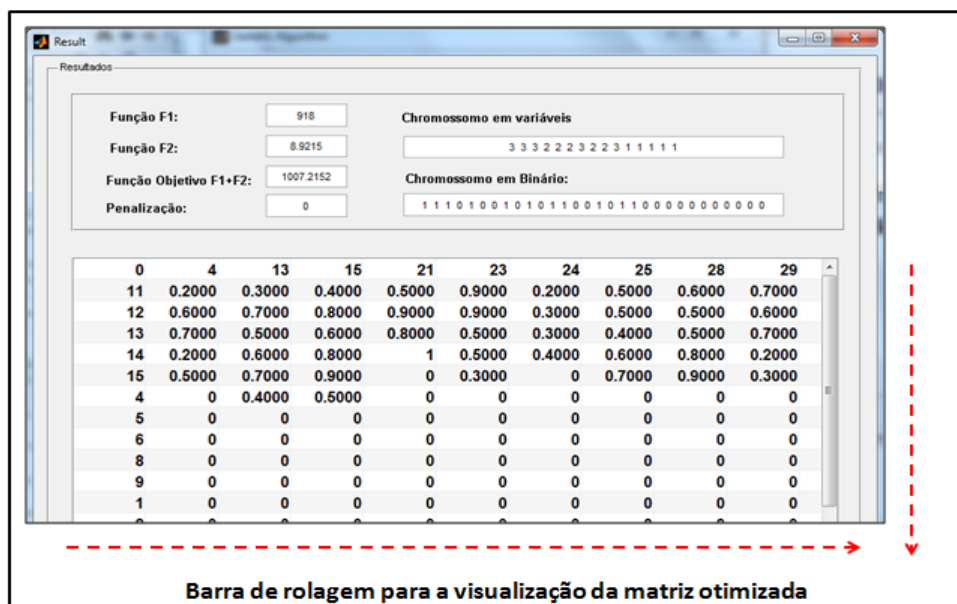
Assim, com o programa inicializado, é exibida a interface de monitoramento do AG durante a iteração do algoritmo, onde são fornecidas informações, conforme comentado no item 3.8.





**Figura. 3.22 - Interface de monitoramento do AG**

No final da simulação do aplicativo é impressa uma janela com os resultados obtidos que podem ser visualizados após a otimização. Entre os resultados que podem ser visualizados o cromossomo na forma binária, o cromossomo em bits, a função objetivo, a penalização (caso o individuo fosse penalizado) e a matriz peça-máquina na forma rearranjada, de acordo com a figura 3.23.



**Figura. 3.23 - Interface de resultados do aplicativo**

## 4 RESULTADOS: EXEMPLOS DE APLICAÇÃO DO APLICATIVO PROPOSTO

Para a validação do aplicativo proposto, este trabalho analisa resultados de matrizes otimizadas (A,B,C), que foram implementados por diferentes métodos, com intuito de comparar a funcionalidade do aplicativo proposto.

### 4.1 Análise da Matriz A:

Trata-se do método proposto por Gonçalves Filho *et. al.*, 2004, onde inicialmente o problema é definido através de uma matriz composta por 15 máquinas e 30 peças, de forma que cada elemento da matriz consiste no respectivo tempo de fabricação. Na figura 4.1, que é apresentada no trabalho de Vernugopal e Naredran, 1992, observa-se a matriz em sua forma inicial.

		Peças																															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
Máquinas	1			0.3			0.6	0.6	0.2	0.2	0.5	0.7							0.4		0.6												
	2	0.4		0.5			0.7	0.3	0.4	0.3	0.6	0.8							0.9		0.2												
	3	0.6		0.7	0.3		0.2	0.4	0.9	0.6	0.2	0.2						0.4	0.3		0.5												
	4		0.2			0.3									0.4	0.7	0.5	0.6		0.2		0.4		0.4					0.5	0.6			
	5		0.2			0.3								0.4		0.5		0.7		0.8		0.9		0.6					0.8	0.2			
	6		0.8			0.9								1		0.7		0.2		0.3		0.4		0.5					0.6	0.8			
	7	0.8		0.9			0.3	0.5	0.5	0.7	0.3	0.5							0.6		0.9												
	8		1.1			1.2								0.3		0.8		0.3		0.9		0.2		0.3					0.4	0.5			
	9		0.4			0.5								0.6		0.9		0.5		0.6		0.7		0.8					0.9	1			
	10	0.6		0.2			0.3	0.9	0.2	0.3	0.4	0.5							0.6		0.8												
	11	0.3		0.3	0.2									0.3		0.4							0.5		0.9	0.2	0.5			0.6	0.7	0.8	
	12				0.6									0.7		0.8							0.9		0.9	0.3	0.5			0.5	0.6	0.7	
	13				0.7									0.5		0.6							0.8		0.5	0.3	0.4			0.5	0.7	0.8	
	14				0.2									0.6		0.8								1		0.5	0.4	0.6			0.8	0.2	0.8
	15				0.5									0.7		0.9									0.3		0.7			0.9	0.3	0.4	
Demanda		155	150	148	160	144	158	152	155	164	148	140	144	145	162	170	140	156	132	172	164	144	158	155	152	140	166	148	145	144	170		

**Figura. 4.1 - Matriz de incidência peça-máquina na sua forma inicial**

Na aplicação deste exemplo consideraram-se alguns parâmetros fundamentais para a implementação deste problema, que são: as características para a formação de células e as variáveis de entrada do AG. As tabelas 4.1 e 4.2 descrevem os parâmetros utilizados para a otimização deste problema, considerando dados operacionais e considerações de projeto da referência bibliográfica citada.

**Tabela 4.1 Parâmetros dos AG para teste no sistema**

Parâmetros AG	
População inicial	800
Taxa de Mutação	0,1
Taxa de <i>Crossover</i>	0,85
Número de gerações	200

**Tabela 4.2 Parâmetros para a formação de células de manufatura**

Parâmetros para formação de células	
Número de máquinas	15
Número de peças	30
Número de células	3
Matriz de incidência peça-máquina	Conforme Figura 4.1
Demanda de peças	Conforme Figura 4.1
Tempo disponível de máquina	147 horas, considerando uma semana para $i$ máquinas

Na figura. 4.2 são apresentados os resultados obtidos pelo aplicativo relacionado aos AG, para o exemplo em análise.

Função F1:	918	<b>Chromossomo em variáveis</b>
Função F2:	8.9215	3 3 3 2 2 2 3 2 2 3 1 1 1 1 1
Função Objetivo F1+F2:	1007.2152	<b>Chromossomo em Binário:</b>
Penalização:	0	1 1 1 0 1 0 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0

**Figura. 4.2 Resultados obtidos relacionados a análise dos AG**

Como pode-se observar, o valor da função objetivo é de 1007.21, onde os valores da função F1 e F2 são de 918 e 8.92 respectivamente. Com os resultados obtidos verifica-se que estes valores são próximos aos dados obtidos no trabalho analisado (Matriz A). Outro ponto a se considerar é que o valor da penalização é igual a zero. Isto significa que o indivíduo selecionado cumpre os requisitos para a sua formação, conforme demonstrado no item 3.4.3. Além disso, o aplicativo apresenta as respostas dos cromossomos expressas em números inteiros (células) ou binário. De posse do melhor indivíduo é possível determinar os grupos de

máquinas que compõem cada célula e suas respectivas famílias de peças. A figura 4.3, que é igual a resposta apresentada por Gonçalves Filho *et. al.*, 2004, ilustra o resultado obtido após a iteração do aplicativo proposto.

		Peças																															
		1	3	6	7	8	9	10	11	17	19	2	5	12	14	16	18	20	22	26	27	4	13	15	21	23	24	25	28	29	30		
Máquinas	1		0.3	0.6	0.6	0.2	0.2	0.5	0.7	0.4	0.6																						
	2	0.4	0.5	0.7	0.3	0.4	0.3	0.6	0.8	0.9	0.2																						
	3	0.6	0.7	0.2	0.4	0.9	0.6	0.2	0.2	0.3	0.5					0.4								0.3									
	7	0.8	0.9	0.3	0.5	0.5	0.7	0.3	0.5	0.6	0.9																						
	10	0.6	0.2	0.3	0.9	0.2	0.3	0.4	0.5	0.6	0.8																						
	4											0.2	0.3		0.7	0.6	0.2	0.4	0.4	0.5	0.6		0.4	0.5									
	5											0.2	0.3	0.4	0.5	0.7	0.8	0.9	0.6	0.8	0.2												
	6											0.8	0.9	1	0.7	0.2	0.3	0.4	0.5	0.6	0.8												
	8											1.1	1.2	0.3	0.8	0.3	0.9	0.2	0.3	0.4	0.5												
	9											0.4	0.5	0.6	0.9	0.5	0.6	0.7	0.8	0.9	1												
	11	0.3	0.3																					0.2	0.3	0.4	0.5	0.9	0.2	0.5	0.6	0.7	0.8
	12																							0.6	0.7	0.8	0.9	0.9	0.3	0.5	0.5	0.6	0.7
	13																							0.7	0.5	0.6	0.8	0.5	0.3	0.4	0.5	0.7	0.8
	14																							0.2	0.6	0.8	1	0.5	0.4	0.6	0.8	0.2	0.8
	15																							0.5	0.7	0.9		0.3		0.7	0.9	0.3	0.4
Demanda		155	148	158	152	155	164	148	140	156	172	150	144	144	162	140	132	164	158	166	148	160	145	170	144	155	152	140	145	144	170		

Figura. 4.3 Solução do problema proposto

#### 4.2 Análise da Matriz B:

Esta análise está relacionada na resolução de um problema, proposto por Leal *et. al.*, 2000, implementado através do método ROC, para os dados ilustrados na matriz indicada na figura 4.4.

		MÁQUINAS								
		M1	M2	M3	M4	M5	M6	M7	M8	M9
P E Ç A S	P1	1				1		1		1
	P2		1				1		1	
	P3			1	1			1		
	P4			1	1					
	P5					1				1
	P6		1				1		1	
	P7	1				1		1		1
	P8				1			1		
	P9	1	1				1		1	
	P10	1	1				1		1	
	P11						1		1	
	P12		1						1	

Figura. 4.4 Matriz de incidência peça-máquina na forma inicial

As tabelas (4.3 e 4.4) ilustram os parâmetros utilizados para a implementação do problema no aplicativo proposto.

**Tabela 4.3 Parâmetros dos AG para teste no sistema**

Parâmetros AG	
População inicial	800
Taxa de Mutação	0,1
Taxa de <i>Crossover</i>	0,85
Número de gerações	200

**Tabela 4.4 Parâmetros para a formação de células de manufatura**

Parâmetros para formação de células	
Número de máquinas	9
Número de peças	12
Número de células	3
Matriz de incidência peça-máquina	Conforme Figura 4.4
Demanda de peças	150 peças, considerando $i$ máquinas
Tempo disponível de máquina	147 horas, considerando uma semana para $i$ máquinas

Para a implementação da análise da matriz B, não foi necessário realizar modificações nos parâmetros de otimização indicados na tabela 4.3. Isto se deve em função de que a matriz B proposta nesta análise possui dimensões menores (número de peças e máquinas) que a matriz proposta na análise da matriz A. Assim, para a otimização do problema, as informações do presente exemplo são menores e mais fáceis de serem tratadas, o que justifica a utilização deste método, utilizando os parâmetros do exemplo anterior.

A resposta obtida neste aplicativo é similar ao resultado encontrado no trabalho de Leal *et. al.*, 2000, porém existem algumas diferenças relacionadas à formação do grupo de máquinas e suas respectivas famílias de peças. A figura 4.5 ilustra a análise comparativa entre os resultados obtidos no trabalho analisado e com o aplicativo proposto.

		MÁQUINAS									
		M4	M3	M7	M9	M5	M1	M6	M8	M2	
P E Ç A S	P3	1	1	1							
	P4	1	1								
	P8	1		1							
	P7			1	1	1	1				
	P1			1	1	1	1				
	P5				1	1					
	P9						1	1	1	1	
	P10						1	1	1	1	
	P6							1	1	1	
	P2								1	1	1
	P11									1	1
	P12										1

		MÁQUINAS									
		M4	M3	M9	M5	M1	M7	M6	M8	M2	
P E Ç A S	P3	1	1				1				
	P4	1	1								
	P8	1					1				
	P7			1	1	1	1				
	P1			1	1	1	1				
	P5			1	1						
	P9						1	1	1	1	
	P10						1	1	1	1	
	P6							1	1	1	
	P2								1	1	1
	P11								1	1	
	P12									1	1

(a)

(b)

**Figura. 4.5 - Comparativo da resposta obtida no trabalho analisado (a) e o aplicativo (b)**

Como pode-se observar, existem pequenas diferenças entre a resposta obtida no trabalho de Leal *et. al.*, 2000 e o aplicativo proposto, como o arranjo dos grupos de máquinas, isto é, a máquina 7 não pertence à mesma célula que foi apresentada no exemplo analisado, logo as famílias de peças também acabam sendo influenciadas. No entanto, pode-se dizer que a resposta obtida pelo algoritmo proposto é similar ao exemplo analisado, pois existem células que possuem as mesmas características, ou as diferenças existentes são aceitáveis do ponto de vista produtivo, ou seja, a diferença existente não representa o sobrecarregamento de uma das três células geradas pelo aplicativo, sendo necessária uma análise mais precisa do usuário para definir qual das configurações obtidas são mais apropriadas.

A figura. 4.6 ilustra os resultados obtidos pelo aplicativo proposto para o exemplo em teste.

Função F1:	600	Chromossomo em variáveis	
Função F2:	6.0739		3 1 2 2 3 1 3 1 3
Função Objetivo F1+F2:	660.7386	Chromossomo em Binário:	
Penalização:	0		1 1 0 0 1 0 1 1 1 1 0 0 1 0 0 0 1 1

**Figura. 4.6 Resultados obtidos relacionados a análise dos AG - Exemplo 2**

Como pode-se observar, o valor da função objetivo é de 660,73, onde os valores da função F1 e F2 são de 600 e 6.07 respectivamente. Com estes resultados pode-se dizer que estes valores são relativamente menores, comparados com a análise da matriz A, ao fato de que a matriz analisada (Matriz B) é mais simples e com características melhores para a otimização.

### 4.3 Análise da Matriz C:

Outro estudo comparativo testado neste trabalho é uma aplicação de Burbidge, que consiste em um método clássico de agrupamento, e é apresentada no trabalho de Chan *et. al.*, 1982, para as informações da matriz ilustrada na figura. 4.7.

M/C NO	COMP. NO.																																																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43												
1																																																							
2		X																																																					
3																																																							
4																																																							
5																																																							
6		X	X																																																				
7		X	X																																																				
8		X	X	X																																																			
9		X	X																																																				
10		X	X																																																				
11																																																							
12																																																							
13																																																							
14		X	X																																																				
15																																																							
16		X																																																					

**Figura. 4.7 Matriz de incidência peça-máquina na sua forma inicial**

As tabelas 4.5 e 4.6 ilustram os parâmetros utilizados para a otimização deste problema.

**Tabela 4.5 Parâmetros dos AG para a otimização do sistema**

Parâmetros AG	
População inicial	6000
Taxa de Mutação	0,1
Taxa de <i>Crossover</i>	0,9
Número de gerações	500

**Tabela 4.6 Parâmetros para a formação de células de manufatura**

Parâmetros para formação de células	
Número de máquinas	16
Número de peças	43
Número de células	5
Matriz de incidência peça-máquina	Conforme Figura 4.7
Demanda de peças	150 peças, considerando <i>i</i> máquinas
Tempo disponível de máquina	147 horas, considerando uma semana para <i>i</i> máquinas

Na análise da matriz C, foram alterados alguns parâmetros para a otimização do sistema, tais como, os valores da população inicial, a taxa de *crossover* e o aumento no número de gerações. Para garantir a melhor combinação de máquinas-células (representação cromossômica), de acordo com o cromossomo gerado, o aumento do número de indivíduos da população inicial tornou-se necessário. A taxa de *crossover* também foi modificada para que os elementos de uma determinada geração, submetido pelo operador genético de *crossover*, tenham uma probabilidade maior de serem adaptados. O número de gerações modificou-se porque neste exemplo de aplicação procurou-se realizar mais iterações, com o intuito de analisar um número maior de indivíduos, considerando as populações iniciais geradas, os processos de seleção e os operadores genéticos aplicados.

A resposta obtida neste aplicativo é similar ao resultado encontrado no trabalho analisado, porém existem algumas diferenças relacionadas a formação do grupo de máquinas e suas respectivas famílias de peças. As figuras 4.8 e 4.9 permitem comparar entre o resultado obtido no trabalho analisado e com o aplicativo proposto.

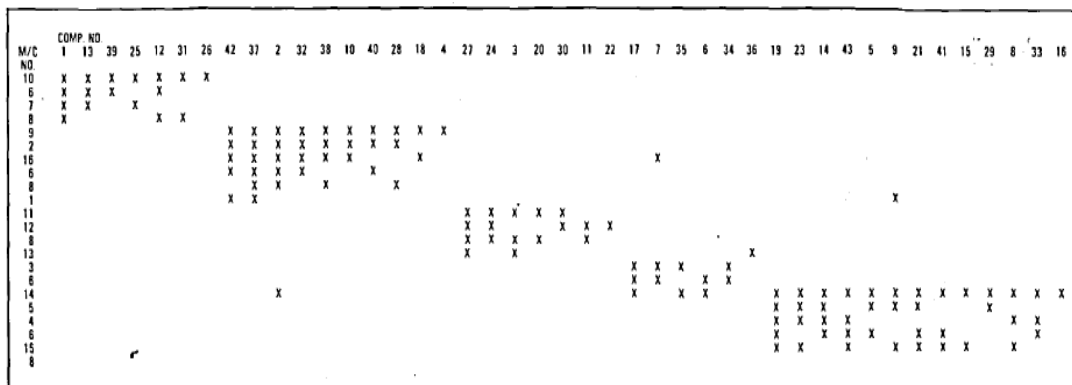


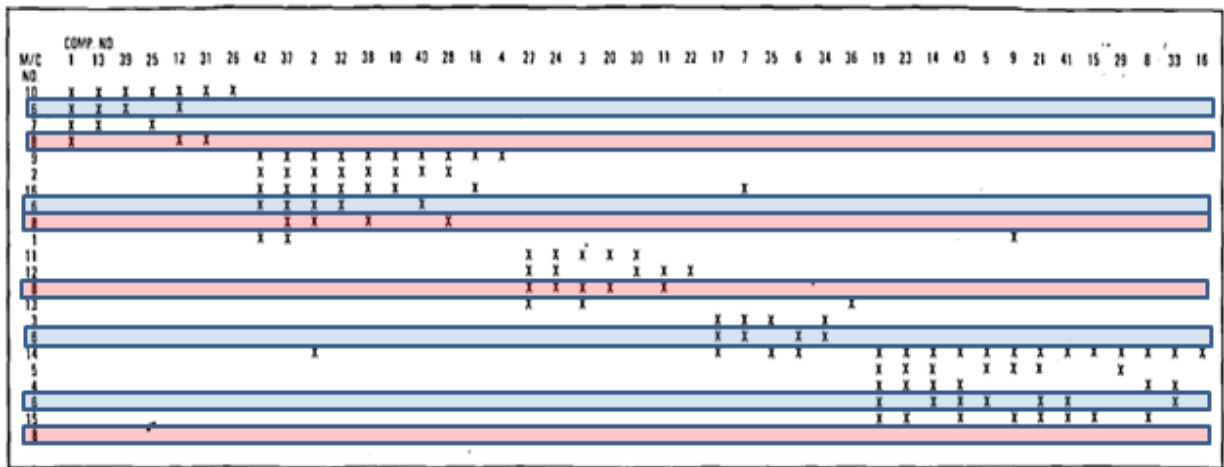
Figura. 4.8 Resultados obtidos por Burbridge [Chan et. al.,1982]

	AG 1	1	13	39	25	12	31	26	42	37	2	32	38	10	40	28	18	4	17	7	35	6	34	19	23	14	43	5	9	21	41	15	29	8	33	16	27	24	3	20	30	11	22	36																
M7	1	1			1																																																							
M10	1	1	1	1	1	1	1	1																																																				
M1										1	1																																																	
M2										1	1	1	1	1	1	1	1																																											
M9										1	1	1	1	1	1	1	1	1	1																																									
M16										1	1	1	1	1	1	1			1																																									
M3																					1	1	1	1																																				
M14																					1	1	1	1																																				
M4																											1	1	1	1																														
M5																										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						
M6																										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						
M8																										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						
M11																																																												
M15																										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						
M12																																																												
M13																																																												

Figura. 4.9 Resultados obtidos com o aplicativo proposto

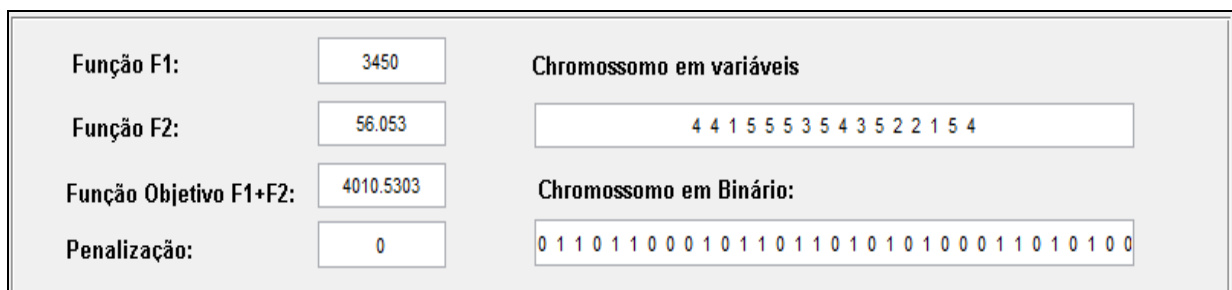


As diferenças existentes entre a resposta obtida no trabalho analisado e o aplicativo proposto estão relacionadas ao arranjo dos grupos de máquinas e às correspondentes famílias de peças respectivamente. Isto pode-se observar no caso da máquina 11, que não pertence a mesma célula que foi apresentada no exemplo de Burbidge. Além disso, observa-se que as máquinas 6 e 8 atendem uma variedade muito alta de peças, fazendo com que um determinado conjunto de peças não esteja relacionada a nenhuma célula formada. Para solucionar este problema, Burbidge, conforme apresentado no trabalho de Chan *et. al.*,1982, adicionou as máquinas 6 e 8 em todas as células formadas, visando atender todas as peças propostas na matriz inicial, conforme figura 4.10.



**Fig. 4.10 Resultado proposto por Burbidge [Chan *et. al.*,1982]**

A figura. 4.11 ilustra os resultados obtidos pelo aplicativo proposto para o exemplo em analisado.



**Fig. 4.11 Resultados obtidos relacionados a análise dos AG - Exemplo 3**

Como pode-se observar, o valor da função objetivo é de 4010.53, onde os valores da função F1 e F2 são de 3450 e 56.05 respectivamente. Com estes resultados pode-se dizer que estes valores são expressivamente maiores, comparados com os exemplos 1 e 2, em função da matriz analisada ser mais complexa e com características que não auxiliam na otimização do problema, como o elevado número de peças que as máquinas 6 e 8 processam.

#### 4.4 Análise comparativa através da variação do número de células

Esta análise visa avaliar a influência na variação do número de células na resposta do algoritmo, conforme ilustra a tabela 4.7.

**Tabela. 4.7 Relação entre função objetivo e a variação do número de células**

Número de Células	Análise da Matriz A (AG)	Análise da Matriz B (ROC)	Análise da Matriz C (Burbridge)
2	701,78	416,27	1590,65
3	1007,21	660,73	2253,26
4	2495,89	939,91	3178,84
5	5118,67	11681,23	4010,53

Como pode-se observar, o valor da função objetivo para os 3 exemplos aumentam conforme o número de células. Desta forma, analisou-se os casos em que os valores da função objetivo foram menores que o valor de referência obtido (valores estabelecidos pela literatura), visando avaliar as características de layout obtidas.

Nesta análise observa-se que existem casos em que a melhor resposta em termos de otimização não é representa a melhor resposta em termos de layout. Isto verifica-se com base nas características obtidas em cada análise.

- **Análise da matriz A – AG com 2 células**

Neste caso, o layout obtido não apresenta as características funcionais de um arranjo físico celular, pois o layout obtido é composto por um elevado grupos de máquinas e famílias de peças

- **Análise da matriz B – ROC com 2 células**

Nesta análise verificou-se que o layout obtido necessita acrescentar máquinas nas células para se obter as características funcionais de um arranjo físico celular, afinal

existem peças com similitudes com máquinas relacionadas as duas células, não tornando esta proposta de arranjo desejável.

- **Análise da matriz C – Burbridge com 2,3 e 4 células**

O layout obtido não representa as características funcionais de um arranjo físico celular, ou seja, não existem características de agrupamento, além de que os grupos de máquinas não apresentam semelhança com a resposta apresentada na literatura.

#### 4.5 Análise comparativa com a literatura

Esta análise estabelece um comparativo entre a resposta obtida pelo algoritmo, considerando a função objetivo, e a resposta obtida pela literatura, conforme ilustra a figura 4.12.

**Tabela 4.8 Análise comparativa entre o aplicativo e a literatura**

Resposta da função objetivo para o aplicativo e a literatura			
	<b>Análise da Matriz A</b>	<b>Análise da Matriz B</b>	<b>Análise da Matriz C</b>
F- Literatura	1007,21	662,47	4094,20
F- Aplicativo	1007,21	660,73	4010,53

Com base nos resultados obtidos, pode-se observar que na análise da matriz A não houveram diferenças entre os valores da função objetivo, pois o método analisado (literatura) possui características similares com o algoritmo proposto. Na análise das matrizes B e C, observa-se que os valores de função objetivo obtidos para o aplicativo foram menores que para literatura. No entanto, cumpre registrar que para realizar está análise, estipulou-se que a resposta obtida pela literatura é a ideal, logo transcreveu-se as matrizes de resposta em forma cromossômica e calculou-se os valores da função objetivo. Além disso, é importante considerar que as diferenças entre as respostas obtidas são pequenas, então com relação as respostas de layout, deve-se analisar qual o tipo de arranjo físico obtido é recomendável para a aplicação desejada.

#### 4.6 Confiabilidade do aplicativo proposto

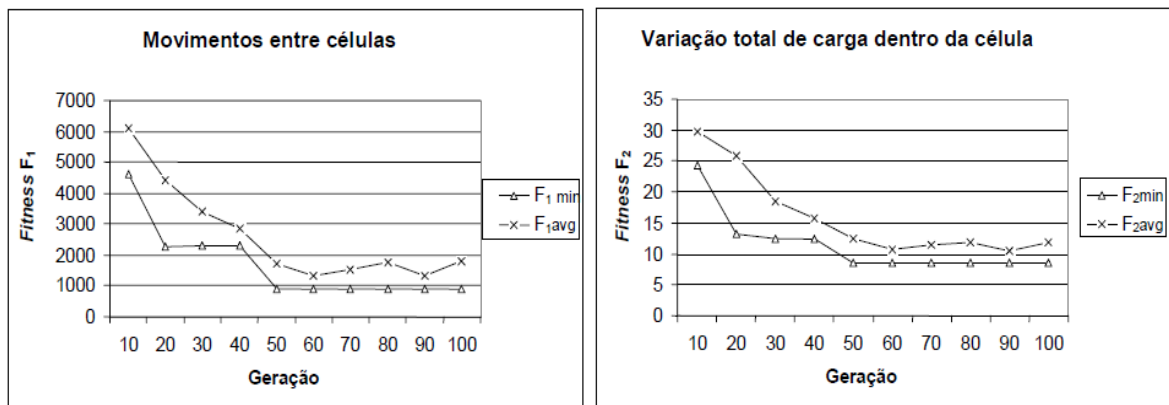
Na implementação de um aplicativo deve-se sempre avaliar a repetitividade dos resultados, visando avaliar sua robustez e confiabilidade. No presente trabalho, para avaliar o algoritmo proposto, analisa-se a exatidão e aleatoriedade de acordo com os valores da função objetivo. A exatidão define-se como o quão próxima a média dos resultados (função objetivo) é dos valores de referência fornecidos na literatura. Já a aleatoriedade consiste no desvio padrão dos resultados, ou seja, está relacionado com a distância entre os resultados obtidos pelo aplicativo e o valor médio.

Desta forma, para realizar esta análise, o aplicativo foi inicializado 15 vezes para as três matrizes analisadas neste capítulo, onde foi verificado o valor da função objetivo e consequentemente os valores de média e desvio padrão.

Com relação aos resultados obtidos, pode-se fazer as seguintes considerações:

- Análise da matriz A:

Nesta análise verifica-se que o algoritmo proposto apresenta uma boa exatidão, pois os valores da função objetivo obtidos pelo algoritmo proposto são de  $F_1=918$  e  $F_2=8.92$ , enquanto os valores fornecidos no trabalho de Gonçalves Filho *et. al.*, 2004 podem ser observados na figura 4.12.



**Fig. 4.13** Relação entre as funções F1 e F2 e o número de gerações

Assim, a média após as 15 inicializações foi de 1165,01, que se pode considerar um valor relativamente próximo, comparado com o valor de 1007,21, que consiste na soma de  $F_1$  e  $F_2$ , de acordo com a Equação 3.3. Com relação à aleatoriedade, observa-se que o desvio obtido após as 15 inicializações foi de 416,33, que comparado com a ordem de grandeza da função obtida é relativamente alta. No

entanto, é importante considerar que o valor da função objetivo foi diferente dos valores de referência somente na segunda e sexta inicialização, onde os valores foram aproximadamente o dobro do resultado esperado.

- Análise da Matriz B:

No caso do trabalho de Leal *et. al.*, 2000 não se pode afirmar que neste exemplo o algoritmo proposto apresenta uma boa exatidão, pois o método utilizado para a solução do problema foi o ROC, que não fornece o valor da função objetivo. Porém, o resultado obtido pelo algoritmo proposto foi de 660,73 (Função objetivo minimizada), que consiste em um valor relativamente baixo, comparado com as demais matrizes analisadas. Com relação à aleatoriedade observa-se que o desvio obtido após as 15 inicializações é igual a zero, o que significa que não há uma distância entre os resultados obtidos e o valor médio.

- Análise da matriz C:

Da mesma maneira que a análise da matriz B, no trabalho de Chan *et. al.*, 1982 não se pode afirmar que neste caso o algoritmo proposto apresenta uma boa exatidão, pois o método utilizado para a solução do problema foi o DCA, que não fornece o valor da função objetivo. O resultado obtido pelo algoritmo proposto foi de 3666,38 (Função objetivo minimizada), que consiste em um valor relativamente alto, comparado com os demais exemplos analisados, em função do elevado número de peças e máquinas fornecidas na matriz proposta por Burbridge. Além disso, os resultados mostram que após as 15 inicializações a média obtida para a função objetivo é de 3855,26, que representa um valor relativamente próximo ao valor da função objetivo minimizada, porém, como comentado anteriormente, não pode-se determinar a exatidão na aplicação neste exemplo. Com relação à aleatoriedade observa-se que o desvio obtido após as 15 inicializações foi de 177,16, que comparado com a ordem de grandeza da função obtida é relativamente baixo. No entanto, nesta análise é importante considerar que a melhor resposta relacionada a diagonalização da matriz (formação de células) não está relacionado consiste no valor de 3855,26, mas sim de 4010,53, ou seja, nesta análise verificou-se que a melhor solução otimizada apresentada no algoritmo não corresponde a melhor proposta de layout.

## 5 CONCLUSÕES

Considerando-se que um modelo de layout celular é o mais indicado quando analisa-se o trabalho em um ambiente fabril em que a produção é feita em pequenos lotes com uma elevada diversidade de produtos, espera-se que o presente trabalho venha fornecer para o usuário um aplicativo para auxiliar no agrupamento das máquinas em células e suas respectivas peças em famílias, com intuito de fornecer uma ferramenta de análise, importante nas etapas de projeto de layout.

Dentre os diversos métodos que são utilizados no planejamento de layout, os métodos heurísticos vem sendo amplamente estudados. Neste contexto, o presente trabalho implementou um método heurístico clássico, como os AG, em uma ferramenta convencional como o *Matlab*. Esta metodologia é válida tanto para o projeto de novos sistemas como para os casos em que deseja-se uma otimização de layout existente, em função de uma determinada demanda de mercado, ou através de uma análise preliminar do fluxo de produção de um ambiente fabril, visando um planejamento racional do layout.

Com relação ao aplicativo desenvolvido, considera-se como uma ferramenta válida no planejamento de sistemas de fabricação, pois auxilia a interação com os parâmetros intrínsecos aos AG, bem como aqueles relacionados ao agrupamento, tais como, o número de células a se formar o número de máquinas, o número de peças e a matriz de incidência peça-máquina no seu arranjo inicial.

Considerando-se a implementação de um algoritmo de otimização, empregando o método dos AG, pode-se inferir as seguintes considerações:

- Em todos os exemplos apresentados pode-se observar que a eficiência dos AG está relacionada diretamente com os parâmetros arbitrados, como tamanho da população, taxa de mutação, taxa de *crossover* e número de gerações.
- Na análise das matrizes B e C, não foi possível mensurar a exatidão dos métodos analisados, pois os mesmos não foram implementados com Algoritmos Genéticos, ou seja, não há um valor de função objetivo de referência para realizar uma análise comparativa. No entanto, nestes exemplos verificou-se que ambos possuem uma baixa aleatoriedade, pois os valores de desvio padrão obtidos em ambos os casos são relativamente pequenos com relação à ordem de grandeza da função objetivo analisada. Desta forma, observa-se que o algoritmo proposto possui relativa confiabilidade em seus resultados, pois, no que diz respeito à exatidão, as respostas

obtidas nas análises são proporcionais ao tamanho e às informações (similaridade) da matriz peça-máquina, além disso, o valor ótimo (otimizado) da função objetivo encontrado em cada exemplo é próximo da média dos valores em suas 15 inicializações. Também deve-se considerar que no exemplo 3 o indivíduo ótimo encontrado na otimização do algoritmo não foi a resposta ideal do ponto de vista de fluxo de produção ou Layout Celular, que é apresentada na resposta de burbridge. Logo, quando se utiliza uma ferramenta de análise deve-se sempre ponderar os valores obtidos no algoritmo com a realidade envolvida no problema.

Por fim, conclui-se que, pelos resultados obtidos neste trabalho, o método implementado através de Algoritmo Genético apresenta-se como alternativa para o agrupamento de células, como se pode observar na sua aplicação, robustez e confiabilidade.

## 6 SUGESTÕES PARA TRABALHOS FUTUROS

Para a implementação de trabalhos futuros enumera-se as seguintes sugestões:

- desenvolver outros aplicativos de otimização (agrupamento de células), utilizando outras compilações de funções pré-existentes no *Matlab*, comparando com os resultados obtidos deste trabalho;
- implementar no aplicativo gerado uma metodologia para avaliar a eficiência do agrupamento gerado, realizando transcrevendo a matriz gerada (otimizada) em um ambiente gráfico de um layout industrial;
- aperfeiçoamento da função objetivo do Algoritmo Genético com o propósito de tornar mais abrangente a análise do Algoritmo com relação as variáveis existentes no projeto de layout, por exemplo, classes de máquinas e categorias de peças.
- realizar testes de otimização com o método de Algoritmo Genético, modificando alguns parâmetros, como tipo de seleção, tipo de mutação, tipo de recombinação, que não foram analisados, visando comparar os resultados com os exemplos desenvolvidos neste trabalho.



## REFERÊNCIAS BIBLIOGRÁFICAS

Arkat, J; Abdollahzadeh, H; Ghahve, H. **A new branch and bound algorithm for cell formation problem.** Applied Mathematical Modelling 36, p 5091-5100, 2012.

Argoud, Ana Rita Tiradentes Terra; Gonçalves Filho, Eduardo Vila; Tiberti, Alexandre José. Algoritmo genético de agrupamento para formação de módulos de arranjo físico. Gest. Prod. [online]. 2008, vol.15, n.2, pp. 393-405. ISSN 0104-530X.

Arzi ,Y; Bukchin, J; Masin, M. **An efficiency frontier approach for the design of cellular manufacturing systems in a lumpy demand environment.** European Journal of Operational, p 346-364, 2001

Ávila, S.L. **Algoritmos Genéticos aplicados na otimização de antenas parabólicas,** Dissertação de mestrado, 98.p, UFSC, 2002.

Borges, S.K. **Resolução de Timetabling utilizando Algoritmos Genéticos e Evolução Cooperativa.,** Dissertação de mestrado, 104.p, UFPR, 2003.

Black, J.T. **O Projeto da Fábrica com Futuro.** Bookman, Porto Alegre, 1998;

Brito, E.S. **Aplicativo para Modelamento 3D de Layout Celular com Base em Tecnologia de Grupo.** Dissertação de mestrado, 29.p, UFRGS, 2010.

Burbidge, J.L. **Planejamento e Controle da Produção.** Atlas, São Paulo, 1983.

Chan, H.M.; Milner, D.A. **Direct clustering algorithm for group formation in cellular manufacture.** Journal of manufacturing system, v. 1, ed. 1, p 65-75, 1982.

Coelho, A.S ; Branco, R.M; Rodrigues, G.S. **Aprimoramento da produtividade de sistemas de manufatura e aplicação de Algoritmos Genéticos na solução de problemas de agrupamentos celulares.** Revista Produção Online, Florianópolis, v.6, n.3, p.116, 2006.

Corrêa, H.L.; Corrêa, C.A. **Administração de produção e de operações:** manufatura e serviços: uma abordagem estratégica. 2.ed. Atlas, São Paulo, 2006.

Defersha, F.M; Chen, M. **A linear programming embedded genetic algorithm for an integrated cell formation and lot sizing considering product quality.** European Journal of Operational Research p 46-69, 2008.

FERNANDES, A.M.R. **Inteligência Artificial:** noções gerais. 2 imp. Florianópolis: VisualBooks, 2005.

Filho, E.V.G; Junior, J.H.C.G; Argoud, A.R.T.T. **Algoritmo Genético para formação de células de fabricação.** ENEGEP, Florianópolis, SC, 2004.

GOMES,H.M. Notas de aula MEC 108 - Algoritmos evolutivos e Neurais para a engenharia, Universidade Federal do Rio Grande do Sul - Programa de Pós-Graduação em Engenharia Mecânica, PROMEC, Porto Alegre.

Goldberg, D. E. "**Genetic Algorithms in Search, Optimization and Machine Learning**", Alabama: Addison Wesley, 413p, 1989.

Groover, M.P. **Automação industrial e sistemas de manufatura/** Mikell Groover; tradução Jorge Ritter, Luciana do Amaral Teixeira, Marcos Vieira; revisão técnica José Hamilton Chaves Gorgulho Júnior - 3. ed. - São Paulo: Pearson Prentice Hall, 2011.

Guerra,C. **Otimização Paramétrica de estruturas treliçadas por Algoritmos Genéticos.** Dissertação de Mestrado, UFRGS, 2008

Kamal, S., Burke, L.I. **FACT: a new network-based clustering algorithm for group technology.** International Journal of Production Research, v. 34, n.4, p.919-946, 1996.

King, J. R. Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm, **International Journal of Production Research**, v.18, n.2, p.214, 1980

Lorenzatto, J.T; Ribeiro, J.L.D. **Projeto de Layout alinhado às praticas de produção enxuta em uma empresa siderúrgica de grande porte**. ENEGEP, Foz do Iguaçu, PR, 2007.

Leal, R. A. ; Correia Filho, A. N. ; Maestrelli, N. C. ; Batocchio, A. . **Uma aplicação do algoritmo ROC (Rank Order Clustering) em manufatura celular**. In: Congresso Nacional de Engenharia Mecânica, 2000, NATAL, RN. Anais do Congresso Nacional de Engenharia Mecânica - CONEM 2000.

Lorini, F.J. **Tecnologia de grupo e organização da manufatura**. Editora UFSC, Florianópolis, 1993.

Mitchell, M. **An Introduction to Genetic Algorithms**. A Bradford book, Massachusetts, 1999.

Muther, R. **Planejamento do Layout: Sistema SLP**. Edgard Blucher, São Paulo, 1973.

Rao, S.S. **Engineering Optimization: Theory and Practice**, John Wiley and Sons, v.4, Hoboken, New Jersey, 2009.

Sayadi, M.K; Ashkan, H. Naini, S.G.J. **Firefly- inspired algorithm for discrete optimization problems: An application to manufacturing cell formation**. Journal of Manufacturing Systems, p 78-84, 2013.

Silva, N.R.S. **Detecção de dano estrutural por Algoritmos Genéticos e Sensibilidade Modal**. Dissertação de Mestrado, UFRGS, 2006.

Singh, N.; Rajamani, D. **Cellular Manufacturing Systems: Design, Planning and Control**. Chapman & Hall, London, 1996.

Souza, S.A. **Algoritmos Genéticos Aplicados à Proteção e Estimação de Harmônicos em Sistemas Elétricos de Potência.** Tese de Doutorado. 259.p.,USP, 2008.

SHINGO, Shigeo, O Sistema Toyota de Produção do Ponto de Vista da Engenharia de Produção, Porto Alegre: Bookman,1996.

Slack, N. Brandão, A.B. Corrêa, H. **Administração da produção** edição compacta.São Paulo: Atlas, 1999.

Tahara, C.S. Carvalho, M.M, Filho, E.V.G. Revisão das Técnicas para Formação de Células de Manufatura. ENEGEP, 1997.

Trindade, A.R. Ochi, L.S. **Um Algoritmo Evolutivo Híbrido para a formação de Células de manufatura.** Pesquisa Operacional, v.26, n.2, p.255-294, Maio a Agosto de 2006.

Vernugopal, V. Narendran, T.T. A genetic algorithm approach to the machine-component grouping problem with multiple objectives. *Computers in Industrial Engineering.* Vol 22, n.4.p.469-480.

Xambre, A.R; Vilarinho, P.M. **A simulated annealing approach for manufacturing cell formation with multiple identical machines.** European Journal of Operational Research, p 434-446, 2003.

Tabela A.2 - Modelo de tabela no modo paisagem

Data	$\alpha_{\text{simulado}}$	$\alpha_{\text{medido}}$	$\beta_{\text{simulado}}$	$\beta_{\text{medido}}$	$\omega_{\text{simulado}}$	$\omega_{\text{medido}}$	O <sub>3</sub>	CO <sub>2</sub>
01/03//2006	0,12	0,13	0,92	0,90	1,55	1,85	270	337
03/03//2006	0,14	0,12	0,92	0,93	1,70	1,90	270	337
04/03//2006	0,11	0,11	0,95	0,98	1,20	1,00	270	337
06/03//2006	0,12	0,12	1,10	1,02	1,45	1,44	270	337
07/03//2006	0,15	0,14	1,22	1,15	1,65	1,70	270	337
08/03//2006	0,17	0,18	0,82	0,84	2,85	2,70	270	337
09/03//2006	0,09	0,10	0,89	0,91	2,65	2,50	270	337
11/03//2006	0,11	0,12	1,12	0,99	3,15	3,45	270	337
13/03//2006	0,11	0,12	1,02	1,05	4,35	4,20	270	337
15/03//2006	0,15	0,14	1,14	1,12	3,85	3,70	270	337
16/03//2006	0,10	0,11	0,88	0,86	3,65	3,60	270	337
17/03//2006	0,11	0,12	0,90	0,91	2,15	2,00	270	337
19/03//2006	0,13	0,11	0,91	0,94	1,45	1,55	270	337
22/03//2006	0,18	0,17	0,99	0,97	3,95	4,00	270	337
24/03//2006	0,17	0,19	0,97	0,96	2,25	2,40	270	337
25/03//2006	0,13	0,14	0,90	0,93	1,75	1,65	270	337
27/03//2006	0,11	0,10	0,88	0,85	1,35	1,30	270	337