



UNIVERSIDADE FEDERAL DO PARANÁ  
PROGRAMA DE PÓS-GRADUAÇÃO DE MÉTODOS  
NUMÉRICOS EM ENGENHARIA

CLAUDIA FRANCESCHETTE

UMA ABORDAGEM HEURÍSTICA PARA O  
PROBLEMA DE CORTE GUILHOTINADO  
BIDIMENSIONAL APLICADO A UMA SITUAÇÃO  
DE CANIBALIZAÇÃO

CURITIBA  
2008

CLAUDIA FRANCESCHETTE

UMA ABORDAGEM HEURÍSTICA PARA O PROBLEMA  
DE CORTE GUILHOTINADO BIDIMENSIONAL APLICADO  
A UMA SITUAÇÃO DE CANIBALIZAÇÃO

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciências, do Programa de Pós-Graduação em Métodos Numéricos em Engenharia, na Área de Concentração em Programação Matemática, dos setores de Ciências Exatas e de Tecnologia da Universidade Federal do Paraná.

Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Neida Maria Patias Volpi

CURITIBA  
2008

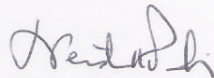
CLAUDIA FRANCESCHETTE

UMA ABORDAGEM HEURÍSTICA PARA O PROBLEMA DE  
CORTE GUILHOTINADO BIDIMENSIONAL APLICADO A UMA  
SITUAÇÃO DE CANIBALIZAÇÃO

Dissertação apresentada como requisito parcial à  
obtenção do grau de Mestre em Ciências, do  
Programa de Pós-Graduação em Métodos  
Numéricos em Engenharia, na Área de  
Concentração em Programação Matemática, dos  
setores de Ciências Exatas e de Tecnologia da  
Universidade Federal do Paraná.

Aprovada em 12 / 11 / 2008

BANCA EXAMINADORA



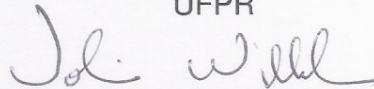
---

Prof.<sup>a</sup>. Dr.<sup>a</sup>. Neida Maria Patias Volpi  
UFPR



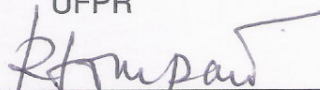
---

Prof. Dr. Celso Carnieri  
UFPR



---

Prof. Dr. Volmir Eugenio Wilhelm  
UFPR



---

Prof. Dr. Raimundo José Borges de Sampaio  
PUC / PR

## AGRADECIMENTOS

À Universidade Federal do Paraná pela oportunidade de realização do Curso de Mestrado em Métodos Numéricos.

À Professora Neida Maria Patias Volpi, pela orientação e dedicação ao longo do desenvolvimento deste trabalho.

A todos os professores do Curso de Métodos Numéricos, que de forma direta ou indireta contribuíram para a minha formação acadêmica.

Ao meu esposo Fernando pela compreensão durante vários momentos ausentes, em que estive dedicando-me a esse trabalho, e, sobretudo pelo amor, carinho e incentivos a continuar os estudos.

Aos meus pais, Cláudio Franceschette e Nair Stecker Franceschette, pela formação, amor e carinho que me proporcionaram durante todos estes anos.

Aos meus amigos e colegas de curso, com quem pude compartilhar muitos conhecimentos, idéias e amizade. Em especial a Vanessa, Karina, Tais, Sandy e Horácio, com quem pude contar em todos os momentos.

À Deus por ter me colocado ao lado de tantas pessoas boas.

## RESUMO

Esta dissertação tem por objetivo apresentar uma aplicação do Algoritmo Genético (AG) para gerar soluções de maneira rápida e eficiente para o problema de corte guilhotinado bidimensional aplicado a uma situação de canibalização. O problema a ser resolvido pode ser definido como: dado um número finito de peças retangulares, de largura  $l_i$ , comprimento  $c_i$  e demanda  $d_i$  a serem obtidos de retângulos maiores de dimensão  $L \times C$  denominados chapas, encontrar padrões de corte da chapa que atendam a demanda de peças solicitadas com o menor desperdício possível. Os AG's são algoritmos de busca baseados em mecanismos de seleção natural (genética), dentro do conceito de "sobrevivência dos mais adaptados". Os operadores genéticos, seleção, crossover e mutação, são aplicados sucessivamente em uma população para criar novas gerações mais adaptadas, ou seja, com melhores resultados para o problema analisado. Para obter a população inicial desenvolve-se uma heurística que gera faixas de peças na chapa. Como resultado deste estudo foi possível obter padrões de corte de chapas, a partir dessa população inicial de faixas, aplicando-se as operações genéticas descritas pelo AG, de forma que a demanda fosse inteiramente atendida com o menor desperdício de material. Mesmo tratando-se de uma abordagem heurística, os resultados da aplicação do AG mostraram-se muito bons quando comparados à solução apresentada por um método exato para o processo de canibalização.

**PALAVRAS-CHAVE:** corte guilhotinado bidimensional; algoritmo genético; canibalização.

## ABSTRACT

This essay aims to present an application of genetic algorithm (AG) to generate quick and efficient solutions to the problem of two-dimensional guillotine cutting applied to cannibalization situation. The problem to be solved can be defined as: considering a finite number of rectangular pieces, with length  $c_i$ , width  $l_i$  and demand  $d_i$  to be obtained from larger-sized rectangles (LxC dimension) called metal sheets, to find appropriate sheet cutting patterns to meet the demand of parts required with the less waste as possible. The AG's are search algorithms based on mechanisms of natural selection (genetic), within the concept of "survival of the most suitable". The genetic operators: selection, crossover and mutation, are applied successively in a population to create new generations more suitable, achieving better results for the problem. To obtain the initial population, a heuristic method is developed to generate strip of pieces on the metal sheet. As a result of this study metal sheet cutting patterns were obtained, from that initial population of strip, applying the genetic operations described by the AG, allowing that the demand could be fully addressed with the least waste of material. Even using an heuristic approach, the results of AG implementation seemed to be very good when compared to the solution applying an accurate method for cannibalization.

**KEYWORDS:** two-dimensional guillotine cutting; genetic algorithm; cannibalization.

# SUMÁRIO

<b>AGRADECIMENTOS</b> .....	<b>i</b>
<b>RESUMO</b> .....	<b>ii</b>
<b>ABSTRACT</b> .....	<b>iii</b>
<b>1. INTRODUÇÃO</b> .....	<b>1</b>
1.1. Apresentação do Problema de Corte Bidimensional.....	1
1.2. Objetivos do Trabalho .....	4
1.2.1. Objetivo geral .....	4
1.2.2. Objetivos específicos .....	4
<b>2. PROBLEMA DE CORTE BIDIMENSIONAL: ALGUNS CONCEITOS</b> .....	<b>5</b>
2.1. Conceitos e Definições.....	5
2.1.1. Classificação dos problemas de corte e empacotamento segundo suas características geométricas.....	6
2.1.2. Problemas de corte em placas.....	9
2.1.3. Revisão bibliográfica .....	17
<b>3. TÉCNICA UTILIZADA: ALGORITMO GENÉTICO</b> .....	<b>24</b>
3.1. Terminologia.....	26
3.2. Principais Aspectos do Algoritmo Genético.....	29
3.2.1. A função objetivo (fitness).....	30
3.2.2. Restrições .....	31
3.2.3. Critérios de parada.....	31



3.2.4. Representação e codificação .....	32
3.2.5. Geração da população inicial .....	33
3.2.6. Avaliação da população .....	35
3.2.7. Processo de seleção .....	35
3.2.7.1. Processo de seleção por torneio .....	35
3.2.7.2. Método da roleta .....	36
3.2.8. Reprodução e cruzamento .....	38
3.2.8.1. Operadores genéticos .....	39
3.2.8.1.1. Crossover.....	39
3.2.8.1.1.1. Crossover em um ponto.....	40
3.2.8.1.1.2. Cruzamento Multipontos .....	41
3.2.8.1.1.3. Cruzamento Uniforme .....	41
3.2.8.1.2. Mutação .....	42
3.2.8.1.3. Elitismo.....	43

#### **4. ALGORITMO GENÉTICO APLICADO A UM PROBLEMA DE CORTE**

<b>BIDIMENSIONAL.....</b>	<b>44</b>
4.1. Descrição do Problema .....	44
4.1.1. Hipóteses .....	46
4.1.2. Geração da população inicial .....	48
4.1.2.1. Dados de entrada do algoritmo genético. Construção de faixas ...	50
4.1.2.2. Descrição do algoritmo de construção de faixas .....	51
4.1.2.3. Descrição do algoritmo de montagem do cromossomo .....	54
4.1.2.4. Descrição do algoritmo genético .....	58
4.2. Resultados....	67

<b>5. CONCLUSÕES</b> .....	<b>79</b>
5.1. Conclusões Gerais.....	79

## LISTA DE TABELAS

Tabela 4.1. Dimensões e demandas.....	48
Tabela 4.2. Saída do Algoritmo Faixas .....	53
Tabela 4.3. Terceira faixa da tabela 4.2 .....	53
Tabela 4.4. Cromossomos População Inicial .....	57
Tabela 4.5. Cromossomos da Vigésima Geração.....	66
Tabela 4.10. Dimensões e demanda.....	67
Tabela 4.11. Faixas .....	68
Tabela 4.12. População inicial .....	69
Tabela 4.13. População da vigésima geração.....	69
Tabela 4..14 Dados de entrada considerando-se as três primeiras peças. ....	73
Tabela 4.15. População inicial para a Tabela 4.14. ....	73
Tabela 4.16. População da quinquagésima geração.....	73
Tabela 4.17. Resultado dos fitness dos 100 testes realizados.....	75

## LISTA DE FIGURAS

Figura 1.1 Corte com desperdício zero. ....	2
Figura 2.1. Objetos em que uma, duas e três dimensões são relevantes.....	7
Figura 2.2. Itens iguais com orientações diferentes.. ....	8
Figura 2.3. Níveis de corte em um padrão.....	10
Figura 2.4. Estágios de corte em um padrão.....	11
Figura 2.5. Exemplo de corte efetuado por guilhotina .....	12
Figura 2.6. Padrão de Corte não Guilhotinado.....	12
Figura 2.7. Estágios de corte com o uso da guilhotina.....	13
Figura 2.8. Desperdício interno e externo .....	14
Figura 2.9. Corte Não Normalizado e (b) Corte Normalizado .....	14
Figura 2.10. Atuação do operador crossover sobre duas chapas .....	19
Figura 2.11. Representação Gráfica do FFD através de exemplo.....	20
Figura 2.12. Resultado final do exemplo prático FFD .....	21
Figura 2.13. Representação Gráfica do FFDH através de desenho .....	22
Figura 3.1. Cromossomo.....	33
Figura 3.2. População Inicial.....	34
Figura 3.3. Método da Roleta.....	37
Figura 3.4. Crossover de um Ponto .....	40
Figura 3.5. Filhos gerados pelos pais 1 e 2 .....	40
Figura 3.6. Cruzamento multipontos .....	41
Figura 3.7. Cruzamento uniforme .....	42
Figura 3.8. Ponto de mutação .....	43
Figura 4.1. Exemplo de Plano de Corte de uma Faixa .....	49

Figura 4.2. Várias faixas formando um padrão de corte da chapa .....	50
Figura 4.3. Padrão de corte da chapa em forma de cromossomo .....	50
Figura 4.4. Representação Geométrica da Terceira Faixa.....	54
Figura 4.5. Ponto de crossover.....	60
Figura 4.6. Cromossomos mutados e com novo fitness .....	60
Figura 4.7. Valor dos fitness de 100 testes realizados. ....	70
Figura 4.8. Solução apresentada pelo QM. ....	72
Figura 4.9. Gráfico da média dos fitness. ....	76
Figura 4.10. Gráfico dos melhores fitness. ....	76
Figura 4.11 Resultado do método exato no QM .....	77
Figura 4.12. Gráfico com os melhores fitness da Tabela 4.17 .....	77
Figura 4.13. Influência do número de gerações na melhoria do fitness .....	78

## LISTA DE FLUXOGRAMAS

Fluxograma 4.1. Algoritmo FAIXAS .....	52
Fluxograma 4.2. Algoritmo TESTE .....	56
Fluxograma 4.3. Operadores Genéticos .....	58
Fluxograma 4.4. Fluxograma Explicativo do Algoritmo SORTEIO_PAIS .....	59
Fluxograma 4.5. Fluxograma Explicativo do Algoritmo CRUZAMENTO .....	61
Fluxograma 4.6. Fluxograma Explicativo do Algoritmo MUTAÇÃO .....	63
Fluxograma 4.7. Fluxograma Explicativo do Algoritmo NOVA .....	65

# 1. Introdução

## 1.1. Apresentação do Problema de Corte Bidimensional

Devido a aspectos econômicos, as indústrias de manufatura vêm-se obrigadas a reduzir os custos para se manterem competitivas. Para isso, medidas como redução no quadro de funcionários, o corte de benefícios e de investimentos não são suficientes e, às vezes, são desnecessários. Neste cenário, para levar vantagem perante as demais, as empresas necessitam tornar seus processos mais rápidos e eficientes.

Existem casos nos quais otimizar um processo de manufatura pode ser impraticável, pois, a busca por essa solução consiste em percorrer um conjunto grande de soluções possíveis. Assim são os problemas de corte bidimensional, comuns em empresas de eletrodomésticos da linha branca, que compram chapas retangulares de aço e precisam cortá-las em itens menores para compor peças que servirão para a montagem de geladeiras, fogões, secadoras de roupas, entre outros. Geralmente, quando são fabricados itens de grande porte, como no caso de geladeiras, as chapas de aço são adquiridas pela fábrica no tamanho exato da peça a ser cortada, atendendo a necessidade de cada produto, ou então de forma que, se a chapa necessite de cortes, a sua dimensão seja pré-calculada para que se tenha desperdício zero. A Figura 1.1 é um exemplo do corte realizado em uma chapa de aço de dimensão 780 mm por 2502

mm, para a obtenção de três peças de dimensão 834 mm por 780 mm, que serão usadas na montagem de um eletrodoméstico.

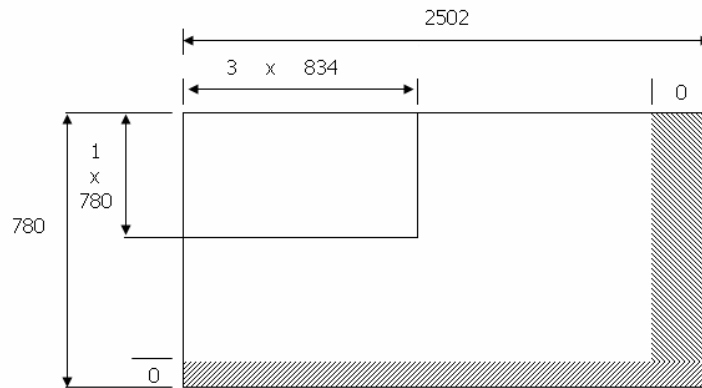


Figura 1.1 Corte com desperdício zero.

Problemas de demanda, gerados principalmente quando um lote de peças é descartado por defeitos no processo de manufatura, implicam em planejar rapidamente a reposição dessas peças perdidas porque nesses casos, faltam chapas com dimensões adequadas para o corte com desperdício zero. Evidentemente que a produção não pode ser parada para aguardar que um novo lote de chapas chegue à fábrica para suprir essa demanda. Então, um procedimento chamado canibalização é utilizado para gerar as peças que foram descartadas. Esse procedimento consiste em utilizar para o corte de algumas peças (somente aquelas que foram descartadas, que normalmente são em número reduzido) uma chapa qualquer, para a qual não foi designada, acarretando assim um desperdício muito grande.

Esta dissertação busca uma abordagem matemática que possa ser aplicada de forma rápida e eficaz na solução deste problema chamado de canibalização. Neste contexto, considere um problema de produção que consiste em cortar peças retangulares em uma chapa de aço, utilizando



guilhotina, de forma a minimizar o desperdício de material. Para tanto, faz-se uso do Algoritmo Genético, que consiste em um modelo computacional inspirado nos mecanismos de evolução de populações de seres vivos, estudado pelo Fisiologista e Naturalista inglês Charles Darwin, desenvolvido por John Holland (1975) e popularizado por David Goldberg (1989).

No desenvolvimento deste trabalho será primeiramente apresentada uma revisão bibliográfica do tema, onde serão abordados alguns conceitos e classificações que influenciam na escolha da técnica a ser utilizada. Essa revisão, que é apresentada no capítulo 02, tem como objetivo, disponibilizar as informações necessárias para a adequada compreensão do método utilizado durante o processo. Serão apresentados, também, estudos realizados por outros autores, que serviram de referência para a construção deste material.

No capítulo 03, será descrito o Algoritmo Genético, método empregado para a resolução do problema, bem como sua terminologia, representação e codificação.

Na seqüência, no capítulo 04, acontecerá o desenvolvimento do algoritmo, bem como sua adequação à situação-problema. Cada um dos sete programas criados para gerar o Algoritmo Genético será comentado e justificado em sua estrutura.

No capítulo 05, serão apresentados os resultados computacionais e suas interpretações, bem como suas análise quantitativa e qualitativa, que podem sofrer pequenas variações cada vez que o processo é realizado, pois a solução inicial é gerada em parte por sorteio. Fechando o trabalho, serão apresentadas as conclusões.

## 1.2. Objetivos do Trabalho.

### 1.2.1. Objetivo Geral.

Melhorar o processo de canibalização das chapas de aço em uma empresa de eletrodomésticos, para reduzir o desperdício desse material.

### 1.2.2. Objetivos Específicos.

São objetivos específicos:

1. Desenvolver uma abordagem matemática do problema de corte bidimensional de chapas, fazendo uso do Algoritmo Genético.
2. Aplicar essa abordagem matemática em um estudo de caso.

## 2. Problema de Corte Bidimensional. Alguns Conceitos.

Apresenta-se neste capítulo conceitos utilizados neste trabalho e algumas abordagens para a solução do problema. O objetivo é familiarizar o leitor com os termos que serão utilizados na abordagem estudada.

### 2.1 Conceito e Definições

Um problema de corte consiste em descobrir o melhor arranjo ou organização de um conjunto de peças de dimensões diversas, que necessita ser projetado em uma chapa com dimensões maiores. Esse é um problema de análise combinatória e, do ponto de vista computacional, é de difícil resolução, pois uma solução deve ser buscada num espaço grande de possíveis soluções, portanto, sendo enquadrado na classe dos NP-hard. Assim, a construção de algoritmos que, apesar de não garantirem soluções ótimas, apresentam soluções de custo reduzido em tempo razoável [Arenales, 1996] [Morabito, 1997].

O problema de corte tem sido largamente estudado nas últimas décadas, com o objetivo de auxiliar as indústrias de couro, madeira, vidro, papelão e aço a minimizar a perda de matéria prima. As indústrias, em função dos custos, buscam diminuir o desperdício de matéria prima, para aumentar o aproveitamento da área ocupada pela demanda de peças que precisam ser produzidas [Fritsch, 1995]. Essas indústrias, normalmente, têm seus processos de cortes automatizados e necessitam um plano de corte sob a matéria prima, ou seja, um planejamento de cada um dos

diagramas contendo a posição de corte das peças.

### 2.1.1 Classificação dos problemas de corte e empacotamento segundo suas características geométricas

Segundo Teodoro [Teodoro, 2003], os problemas de corte e empacotamento podem ser classificados, em geral, por certas características que serão determinantes para a escolha das técnicas ou algoritmos utilizados para resolvê-los. Algumas das principais são:

#### **Dimensionalidade:**

Ao analisar a dimensionalidade de um problema, verificam-se quantas dimensões são realmente relevantes a este problema. Para o caso unidimensional, por exemplo, no corte de bobinas de aço, a largura é a única dimensão tratada. No caso bidimensional, onde se tem o corte de chapas, deve-se considerar a largura e o comprimento dos objetos. Já no caso tridimensional, considera-se também a altura dos objetos e esse problema é comum, por exemplo, para empresas de transporte que procuram fazer o menor número de viagens acomodando as caixas nos caminhões da melhor maneira possível. (Figura 2.1) [Teodoro,2003]

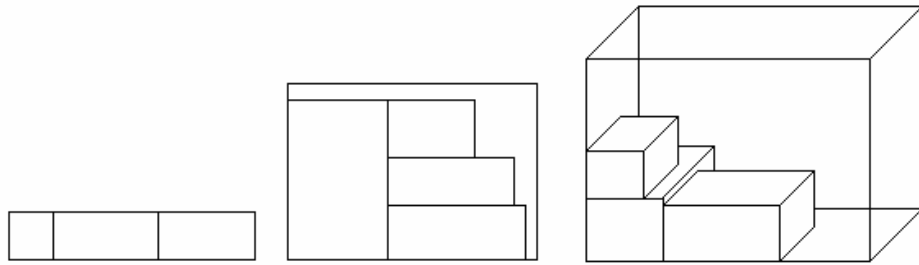


Figura 2.1 Objetos em que uma, duas e três dimensões são relevantes. [Teodoro, 2003]

Tem-se, ainda, o caso multidimensional, quando se pode ter mais de três variáveis relevantes ao processo, neste caso, além das dimensões dos objetos, outras características do problema são consideradas, tais como o tempo disponível para realizar os cortes ou a disponibilidade de maquinário para realizar estes cortes.

### **Demanda**

Em um problema de corte, a demanda representa a quantidade de itens que devem ser obtidos dos objetos maiores, como por exemplo, as peças que deverão ser cortadas em uma chapa de aço ou placa de madeira. Normalmente a variação dos itens é pequena e a demanda é alta.

Um problema é considerado restrito quando a demanda imposta deve ser respeitada, este é o caso do problema que será tratado neste trabalho.

### **Orientação**

Quanto à orientação, as peças e as chapas podem assumir sentidos diferentes sendo fixa ou com rotação (Figura 2.2).

O que define se o problema é orientado ou não, é se o item a ser cortado pode ser girado ou não, como por exemplo, o corte de chapas de madeira, onde deve ser observada a orientação das suas fibras, para garantir uniformidade às peças. Já as chapas de aço não apresentam tal problema.

Ao determinar que os itens possam ser girados, a complexidade do problema aumenta, pois, tem-se que considerar todas as rotações possíveis. Neste trabalho serão consideradas rotações na horizontal e vertical.

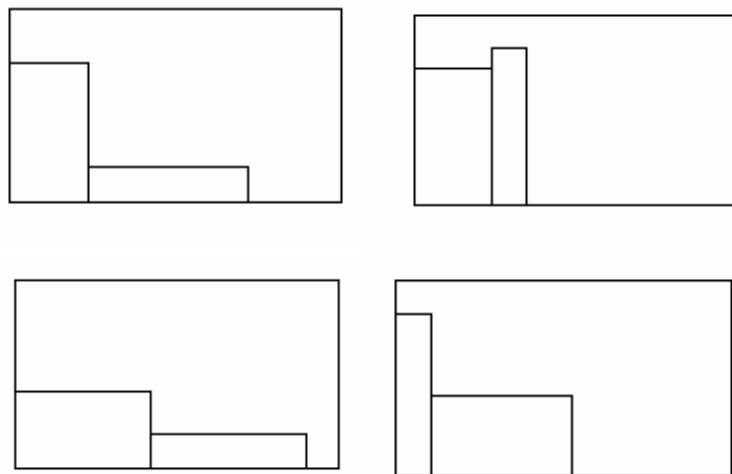


Figura 2.2 Itens iguais com orientações diferentes. [Teodoro, 2003]

### **Disponibilidade:**

A disponibilidade considera no problema a quantidade de objetos e matéria-prima que podem ser usados livremente, sem restrições. No caso deste problema, por estar-se trabalhando justamente com falta de determinada chapa em estoque, considera-se a disponibilidade como fator fundamental para justificar a aplicação deste trabalho.

## 2.1.2. Problema de corte em placas

São apresentadas, a seguir, algumas entidades e conceitos básicos relacionados ao problema de corte. [Silva, 2005]

### **Pecas:**

Chama-se de peças pequenos retângulos [Fritsch, 1995] que são itens de uma demanda de pedidos em uma linha de produção, que devem ser projetados sobre uma lâmina [Daza, 1995] [Oliveira, 1990] denominada aqui de chapa.

No caso deste trabalho, as peças e as chapas são regulares, formadas por retângulos ortogonais [Morabito, 1995]. As peças não regulares possuem formas não convexas, não simétricas, ângulos diferentes, quantidade de lados variados podendo ter retas ou curvas [Fritsch, 1995].

### **Nível de Corte:**

Na indústria podem existir restrições ao procedimento de corte. Uma delas é o nível de corte das máquinas, isto é, quantas vezes uma chapa será cortada, e conseqüentemente a máquina que realiza tais cortes será utilizada, para obterem-se itens desejados. Na Figura 2.3 são apresentados sete níveis de corte, que representam a quantidade de

cortes que serão feitos dentro dessa chapa. Primeiramente, são realizados os cortes de nível 1 e 2, então, a chapa é girada para que os cortes de nível 3, 4, 5 e 6 possam ser realizados. Na seqüência, a chapa é girada novamente para a realização do último corte, de nível 7. Ressalta-se aqui, o tempo gasto para a realização de todo o processo, que apesar de não estar no foco deste trabalho, sempre gera custo para as empresas.

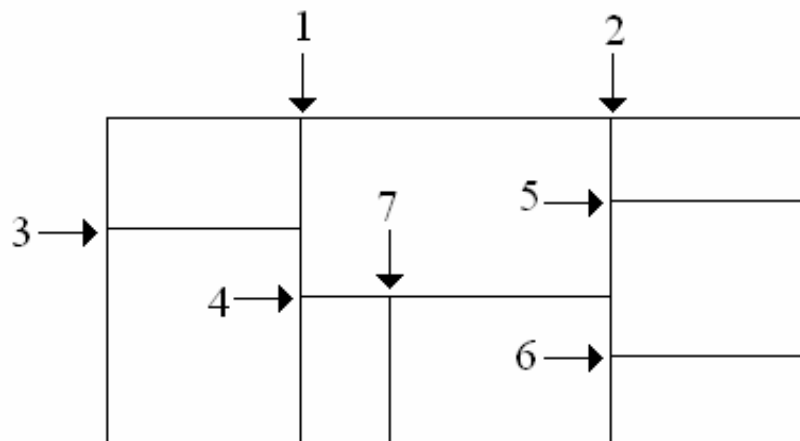


Figura 2.3 Níveis de corte em um padrão.

### **Padrão de Corte:**

O arranjo das peças na chapa é denominado de “plano de corte” e, quando este plano é utilizado mais de uma vez, é denominado de “padrão de corte”. Neste trabalho usam-se tais conceitos como sinônimos.



### **Estágio de Corte:**

As guilhotinas realizam cortes por estágios, isto é, os retângulos são obtidos fazendo uma seqüência de vários cortes em uma mesma direção, por exemplo, horizontais e depois, nas sobras, fazem-se vários cortes verticais e assim por diante, alternando as direções e dessa maneira diminuindo o tempo de utilização da mesma. Em cada estágio a direção dos cortes é ortogonal a do estágio anterior. Na figura 2.4, pode-se perceber dois estágios de corte, um na vertical (que nesse exemplo ocorre primeiro) e outro na horizontal.

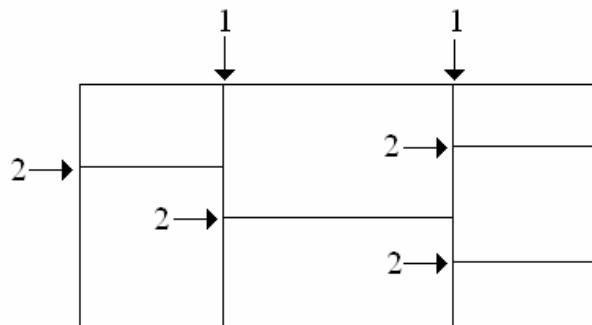


Figura 2.4 Estágios de corte em um padrão. [Silva, 2005]

### **Corte Guilhotina:**

O tipo de corte é considerado guilhotina quando este percorre a placa de um lado ao outro, ou seja, após o corte são produzidos dois retângulos como está representado na Figura 2.5 [Farley, 1990] [Morabito, 1997] [Morabito, 1995] [Morabito, 1996]. Quando o corte é do tipo guilhotina, diz-se que o padrão de corte é do tipo guilhotina.

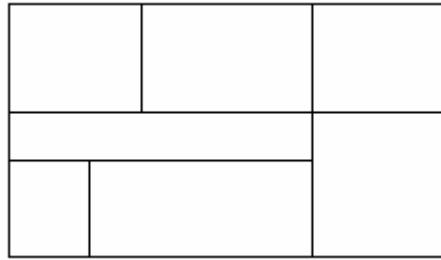


Figura 2.5 Exemplo de corte efetuado por guilhotina.

Na Figura 2.6 o padrão gera cinco retângulos e neste caso o corte é considerado não guilhotina, isso porque, cada um dos cortes não pode percorrer a chapa de um lado a outro.

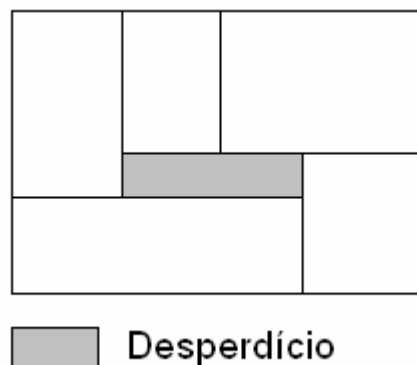


Figura 2.6 Padrão de Corte não Guilhotinado. [Silva, 2005]

No padrão de corte guilhotina, o primeiro corte normalmente é paralelo ao lado mais curto da chapa, para isso, deve-se observar a estrutura da guilhotina. O segundo corte é feito no sentido oposto ao primeiro, representando assim, uma seqüência de estágios sucessivos [Silva, 2005]. Dois fatores definem a prioridade de cada corte, um deles é a largura da guilhotina, o outro é a dimensão das peças que deve ser

respeitada de forma que estas não sejam danificadas durante utilização da guilhotina.

Na Figura 2.7 os números representam os estágios de corte e as setas a incisão da guilhotina.

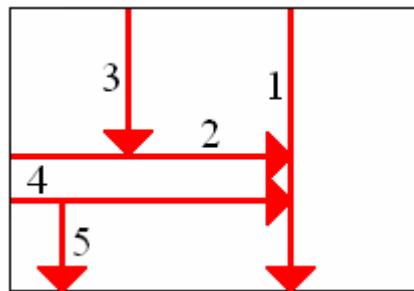


Figura 2.7 Estágios de corte com o uso da guilhotina.

O corte pode ser restrito ou irrestrito. Restrito é quando há um limite em relação ao número de peças de um determinado tipo, e é irrestrito quando não existe este limite. À medida que as peças são combinadas na chapa, entre elas pode haver um espaço não aproveitado, que se chama desperdício interno [Vasko, 1989]. Isso ocorre normalmente em função das dimensões das peças serem diferentes. Caso as peças tenham dimensões iguais, e/ou os lados de uma peça toquem os lados de outras peças de forma a não deixar espaço entre elas, então, o desperdício interno não existe. O desperdício externo ocorre quando o espaço está além do conjunto de peças já projetadas e nele não cabe mais nenhuma outra peça na parte externa da chapa. Figura 2.8.



Figura 2.8 Desperdício interno e externo.

Segundo Christofides [Christofides, 1995] o corte pode ser normalizado ou corte não normalizado. O corte é normalizado quando uma peça é projetada ao lado da outra e tem um de seus lados encostado nesta, ou seja, entre as peças não tem área de desperdício interno, como mostrado na Figura 2.9 (b) .

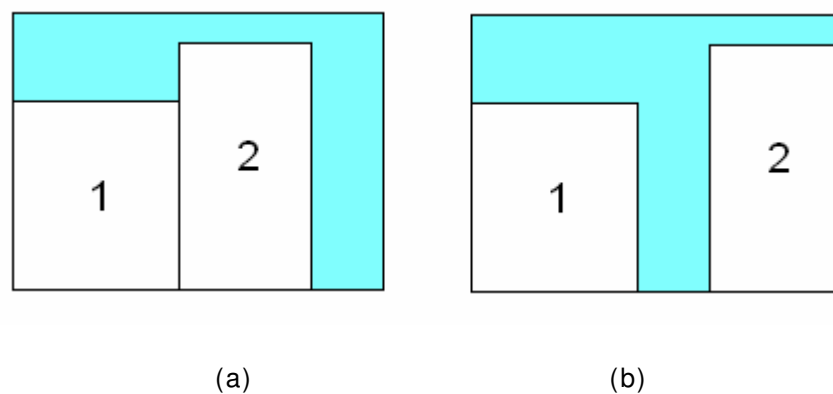


Figura 2.9 (a) Corte Não Normalizado e (b) Corte Normalizado.

Para a utilização de guilhotina, o corte de vê ser ortogonal, ou seja, quando as peças regulares são projetadas, um de seus lados é paralelo a um dos lados da chapa.

O corte ainda pode ser classificado como homogêneo, quando peças a serem projetadas têm as mesmas dimensões e não homogêneo, quando tais peças possuem dimensões diferentes [Araujo, 2006]. Também é preciso que se faça um controle para que a soma da área da próxima peça projetada mais a área das peças já projetadas, não ultrapassem a largura e a altura da chapa, e dessa forma o corte seja possível.

### **Objetivos da resolução de um problema de corte guilhotina:**

Os objetivos de um problema de corte guilhotina estão relacionados ao que se deseja otimizar no problema, dessa forma, o objetivo pode ser maximizar o total da área ocupada, o que é o equivalente a minimizar áreas em desperdício, ou então pode ser maximizar o valor total de pedaços cortados independente da área.

Dessa maneira, um problema de corte guilhotinado consiste em um conjunto de peças retangulares, cada uma com um dado valor e tamanho que devem ser cortadas em uma chapa retangular principal, no qual o objetivo pode ser:

- Maximizar o valor de peças cortadas;
- Minimizar o desperdício nos padrões;
- Determinar a melhor seqüência de cortes;
- Determinar soluções exatas;
- Determinar soluções aproximadas;

Qualquer um destes objetivos pode estar sujeito a restrições quanto ao número de peças, demanda, número de guilhotinas e tempo disponível para o trabalho. Ainda pode-se ter a combinação de dois ou mais objetivos. Este trabalho está concentrado em encontrar padrões de corte de modo a minimizar o desperdício e atender à demanda.

### 2.1.3. Revisão bibliográfica

Trabalhos na área de corte guilhotina bidimensional foram publicados por Gilmore e Gomory [Gilmore, 1963], que primeiramente apresentaram um algoritmo para resolver o problema de corte com uma ou mais dimensões. Em outro artigo dos mesmos autores [Gilmore, 1995] foi apresentada uma solução para o mesmo problema envolvendo corte em estágios. Herz [Herz, 1972] propôs uma solução exata para padrões de corte guilhotina através de uma técnica de árvore recursiva. Wang [Wang, 1983] representou uma solução que gera padrões condicionados por sucessivos cortes horizontais e verticais. Daza, Alvarenga e Diego [Daza, 1995] apresentaram uma melhoria para o algoritmo de Wang, e propuseram uma solução para o Problema de Corte Bidimensional restrito através da associação do algoritmo a um Grafo do tipo “E/Ou” que serve para representar todos os possíveis padrões de corte de uma chapa. Nessa teoria, um problema (P) é subdividido em subproblemas ( $P_1, P_2, P_3, \dots$ ), formando uma árvore (denotada com grafo), em que, para se resolver o problema P, precisa-se primeiramente, resolver os subproblemas  $P_1, P_2, P_3, \dots$ .

Morabito e Arenales [Morabito, 1995] apresentam uma abordagem para o Problema de Corte Guilhotina Bidimensional irrestrito e restrito, não estagiado e estagiado, que utilizava o conceito de Grafo “E/Ou” e o método “branch-and-bound”. Outro trabalho dos mesmos autores, propõe uma solução para corte guilhotina bidimensional para uma grande demanda de variedade de pedidos de peças, que utilizou a técnica de padrões de faixa.

[Morabito 1997].

Carneiro, Alvarenga e Parada [2001] apresentam um procedimento baseado no Algoritmo Genético para o problema de corte bidimensional restrito tipo guilhotina, tendo como referência os operadores genéticos: reprodução, crossover e mutação. Para a formulação de tal problema foi considerado o caso da minimização de perda de material, desejando-se cortar a partir de chapas planas retangulares, mantidas em estoque, um conjunto de peças menores, também retangulares, com formas e quantidades máximas pré-estabelecidas.

Com o propósito de gerar as soluções em forma de chapas, com as disposições dos retângulos a serem cortados e garantir o tipo de corte guilhotina, foi usada a noção do retângulo soma, ou seja, retângulos constituídos a partir de dois outros retângulos menores, segundo um tipo de formação: horizontal ou vertical, as quais são estabelecidas baseadas no conceito de Wang [1983]. Os retângulos soma são formados a partir do Módulo Gerador de Placas (MGP) que toma todas as possibilidades de formações horizontais ou verticais entre os retângulos mantidos em estoque e seleciona as melhores formações possíveis, gerando, assim, os retângulos soma. Numa etapa seguinte, esses retângulos soma podem recombinar-se entre si e com retângulos mantidos em estoque. No caso do problema de corte, o papel desempenhado pelo operador crossover, é o de gerar novas soluções promissoras. A Figura 2.10, retirada do mesmo artigo, ilustra uma forma de atuação do operador crossover sobre as duas chapas.



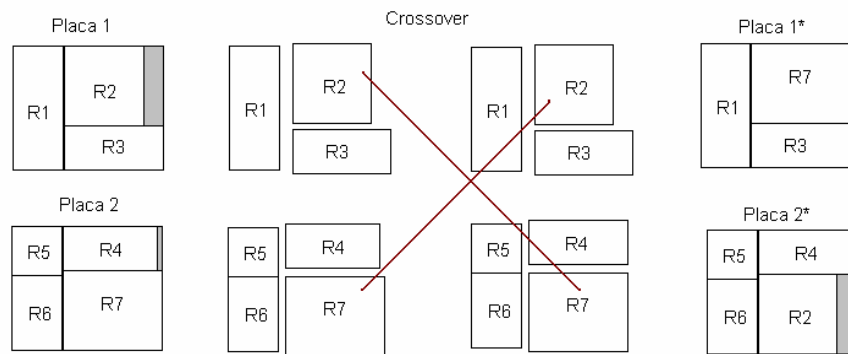


Figura 2.10 Atuação do operador crossover sobre duas chapas.  
Fonte: Alvarenga, 1990

Segundo a Figura 2.10, parece que há uma troca pura e simples entre as configurações de corte após o ponto crossover, mas, na realidade, devido à construção gráfica, a troca não é tão simples, pois, quando um determinado retângulo soma é retirado da formação da configurações de corte, ele causa a quebra de toda formação em que está inserido e, a partir desse momento, as configurações de corte deixam de existir. Esses retângulos soma, juntamente com os retângulos mantidos em estoque, serão a base de formação de novas configurações de corte, que será formada pelo Módulo Gerador de Placas na geração seguinte.

Outro item importante, tratado por Teodoro [Teodoro, 2003], em seu trabalho de corte bidimensional, é a geração do conjunto inicial de padrões de corte. Para gerar de um conjunto inicial de padrões de qualidade, e garantir assim boas soluções para o problema, duas heurísticas foram utilizadas, o FFD e o FFDH, o primeiro também foi utilizado por Arenales [2005], em um estudo de heurísticas para um problema de corte e de estoque com reaproveitamento de sobras.

### **FFD (First Fit Decreasing)**

É um algoritmo de empacotamento unidimensional que consiste, primeiramente, em ordenar os itens de maneira decrescente e posicionar a primeira peça na parte inferior de uma chapa principal (retângulo maior). Fazendo o mesmo para a segunda peça. Se esta peça não couber na primeira chapa principal (logo acima da primeira peça), uma nova chapa é requerida. As peças subseqüentes continuam o processo tentando empacotar as chapas na ordem em que forem requisitadas e usando novas chapas caso não haja espaço nas existentes. Vê-se um exemplo prático, como segue a Figura 2.11 [Teodoro, 2003].

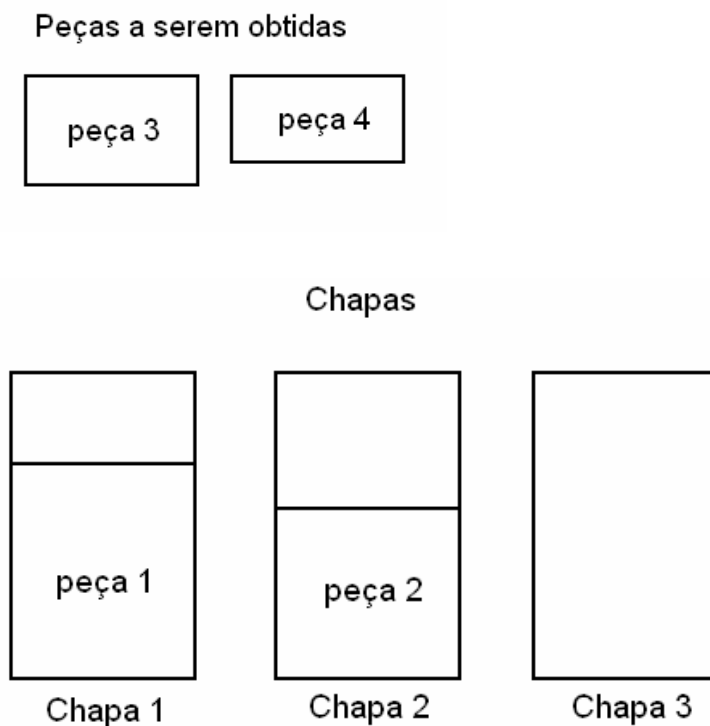


Figura 2.11 Representação Gráfica do FFD através de exemplo.  
Fonte: Teodoro, 2003.

Neste exemplo, a próxima peça a ser empacotada é a peça 3. Pela lógica do FFD, esta peça seria colocada na chapa 2 e a última peça na chapa 1, totalizando dois padrões de corte necessários, como na figura 2.12.

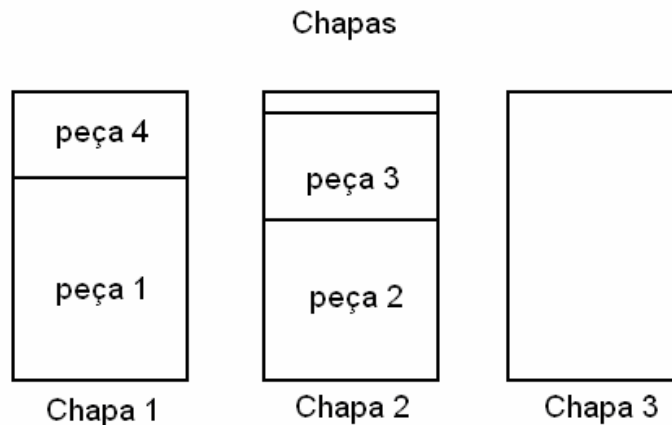


Figura 2.12 Resultado final do exemplo prático FFD.  
Fonte: Teodoro, 2003

### **FFDH (First Fit Decreasing Height)**

O FFDH é um algoritmo de empacotamento em faixa e funciona da seguinte maneira: primeiramente ordena as peças por ordem decrescente de largura e coloca a primeira peça na parte inferior, justificada à esquerda. Pega, então, a segunda peça e a coloca, se possível, ao lado da primeira peça. Se não for possível, traça uma faixa na altura da primeira peça e coloca a segunda peça em cima, alinhada à esquerda. Repete-se o procedimento para as peças subsequentes, tentando posicioná-las nas faixas na ordem em que foram criadas ou criando novas faixas caso não seja possível empacotar essas peças. Vê-se um exemplo prático na figura 2.13 [Teodoro, 2003].

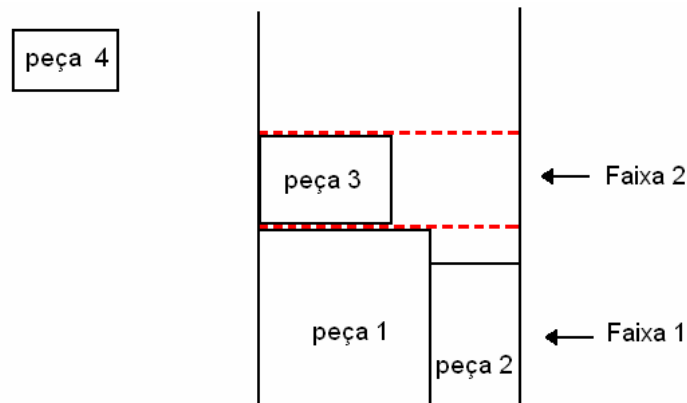


Figura 2.13 Representação Gráfica do FFDH através de desenho.  
Fonte: Teodoro, 2003

Poldi [2006], trata de um problema de corte unidimensional para problemas de baixa demanda onde também é utilizado o FFD. A proposta do trabalho de Poldi é a utilização de heurísticas construtivas (FFD) de aproximação para obtenção de soluções inteiras que superam a dificuldade de heurísticas anteriormente utilizadas, em face de demandas baixas e que necessitam soluções boas e rápidas.

Pileggi [2007] apresentou um estudo sobre o problema de corte guilhotinado. Este consiste em cortar objetos de tamanhos e quantidades conhecidos, para atender uma demanda de itens de tamanhos e quantidades especificados pelos clientes, de forma a, por exemplo, minimizar a perda do material cortado com sequenciamento de padrões de corte (um padrão de corte define uma maneira de arranjar itens dentro de objetos). Dessa forma, deve-se definir uma seqüência em que os padrões serão processados a fim de, por exemplo, minimizar o número máximo de pilhas abertas. Este último problema surge em determinados processos de corte, por exemplo, quando há restrições dos equipamentos de corte,

como ocorre com estações de descarregamento automáticas dos itens cortados em fábricas de chapas duras, ou no caso de limitação de espaço físico para armazenagem dos itens ao redor da máquina de corte. Para a solução do problema foram feitos cortes longitudinais produzindo conjuntos de faixas e a utilização de abordagens heurísticas para resolver o problema combinado de geração e sequenciamento de padrões de corte bidimensional.

Os trabalhos publicados trouxeram várias opções em procedimentos e técnicas, apresentando soluções, heurísticas, modelos e conceitos sobre diferentes enfoques do Problema de Corte Guilhotinado Bidimensional, criando uma grande base de conhecimento para o desenvolvimento deste trabalho. O procedimento de crossover conforme apresentado por Alvarenga [2001] neste capítulo na Figura 2.10, mostrou-se bastante interessante, pois, mesmo não chegando necessariamente a uma solução exata, por tratar-se de uma metaheurística, apresenta inúmeras possibilidades de padrões de corte, justificando, assim, sua escolha como método a ser utilizado.

### 3. Técnica Utilizada: Algoritmo Genético

Os problemas de otimização são baseados nas características:

- i) a codificação do problema;
- ii) a função objetivo que se deseja maximizar ou minimizar;
- iii) e o espaço de soluções associado;

A codificação é a descrição matemática do problema, com os parâmetros e restrições, e, uma saída, que é o valor da função objetivo, que indica se um determinado conjunto de parâmetros é bom ou não para resolver este problema.

Desenvolvido por John Holland em 1960, os algoritmos genéticos (AG) são uma família de métodos computacionais inspirados na evolução natural, que incorporam uma solução potencial para um problema específico numa estrutura semelhante a de um cromossomo e aplicam operadores de seleção e "cross-over" ou recombinação a essas estruturas de forma a preservar informações importantes relativas à solução do problema.

Mitchell [Mitchell, 1998] descreve em seu livro *An Introduction to Genetic Algorithms* o algoritmo criado por Holland, bem como a terminologia utilizada por ele. Dessa forma, assim como as características hereditárias são transmitidas por gerações através de estruturas denominadas genes, as informações relativas à solução do problema também são recombinadas e repassadas através das iterações realizadas pelo AG.

AG's simples normalmente trabalham com um conjunto de soluções possíveis chamadas de cromossomos. Existem três tipos de representação

possíveis para os cromossomos: binária, inteira ou real. De acordo com a classe de problema que se deseja resolver, pode-se usar qualquer um dos três tipos.

Uma implementação de um algoritmo genético começa com uma população aleatória de indivíduos ou cromossomos, cada um com um valor de adaptabilidade associado chamado aptidão, ou fitness. Essas estruturas desenvolvem, através de operações genéticas como cruzamento e mutações, uma nova geração de indivíduos usando os princípios darwinianos de reprodução e sobrevivência dos mais aptos. Cada indivíduo na população representa uma possível solução para um dado problema. Dessa forma, o Algoritmo Genético procura aquela solução que seja muito boa ou a melhor, visando a otimização da função objetivo.

### 3.1. Terminologia

A teoria da evolução diz que o meio ambiente seleciona, em cada geração, os seres vivos mais aptos de uma população. Como resultado, somente os mais aptos conseguem se reproduzir, uma vez que os menos adaptados geralmente são eliminados antes de gerarem descendentes. Durante a reprodução, ocorrem, entre outros, fenômenos como mutação e cruzamentos, que atuam sobre o material genético armazenado nos cromossomos. Estes fenômenos levam à variabilidade dos seres vivos da população. Sobre esta população diversificada age a seleção natural, permitindo a sobrevivência apenas dos seres mais adaptados (Mitchell, 1998).

O fato de o Algoritmo Genético ser baseado na teoria da evolução explica o fato dos AG's possuírem muitos termos utilizados na Biologia. A seguir, apresenta-se a descrição dos principais termos encontrados na literatura:

- **Cromossomo** – Na Biologia, é o conjunto completo de genes de um organismo. No AG, representa a estrutura de dados que codifica uma solução para um problema, ou seja, um cromossomo representa um simples ponto do espaço de busca [Silva, 2005] ;

- **Gen ou gene** – Na Biologia, é a unidade de hereditariedade que é transmitida pelo cromossomo é que controla as características do organismo. Nos AG's, é um parâmetro codificado no cromossomo, ou seja,



um elemento do vetor que representa o cromossomo, podendo ser binário, inteiro ou real [Silva, 2005].

- **Indivíduo:** Um simples membro da população. Nos AG's, um indivíduo é formado pelo cromossomo e sua aptidão ou fitness [Silva, 2005].

- **Genótipo:** Na Biologia, representa a composição genética contida no cromossomo. Nos AG's, representa a informação contida no cromossomo [Silva, 2005].

- **Fenótipo** – Nos AG's, representa o objeto, estrutura ou organismo construído a partir de informações do genótipo. É o cromossomo decodificado. Por exemplo, considere-se que o cromossomo codifica parâmetros como as dimensões dos itens a serem cortados em um problema de corte unidimensional. O fenótipo seria os setups formados para o problema de corte [Silva, 2005];

- **Alelo:** Na Biologia, representa uma das formas alternativas de um gene. Nos AG's, representa os valores que o gene pode assumir. Por exemplo, um gene que representa o parâmetro cor de um objeto poderia ter o alelo azul, preto ou verde [Silva, 2005];

- **Epistalsia:** Interação entre genes e cromossomo, isto é, quando um valor de gene influencia o valor de outro gene. Problemas com alta epistalsia são de difícil solução por AG's [Silva, 2005];

- **População:** Conjuntos de cromossomos ou solução no espaço de busca;

- **Geração:** Iteração completa do AG que gera uma nova população;

- **Operações genéticas:** Operações que o AG executa sobre cada um dos cromossomos;

- **Espaço de busca ou região viável:** É o conjunto, espaço ou região que compreende as soluções possíveis ou viáveis do problema a ser otimizado. Deve ser caracterizado pelas funções de restrição do problema que definem de forma viável o problema a ser resolvido [Silva, 2005];

- **Função objetivo ou de avaliação:** É a função que se quer otimizar. Ela contém a informação numérica do desempenho de cada cromossomo na população (fitness). Nela estão representadas as características do problema que o AG necessita para realizar o seu objetivo, sendo expressa normalmente como  $F(x_1, x_2, x_3, \dots, x_n)$ , onde  $x_1, x_2, x_3, \dots, x_n$  são as variáveis que o algoritmo procura determinar para otimizar  $F$ . Esta função objetivo é em princípio avaliada para cada cromossomo  $C_i$  da população [Silva, 2005].

Deve ser observado que cada cromossomo, chamado de indivíduo no AG, corresponde a um ponto no espaço de soluções do problema de otimização. O processo de solução adotado nos algoritmos genéticos consiste em gerar, através de regras específicas, um grande número de indivíduos, população, de forma a promover uma varredura tão extensa quanto possível do espaço de soluções.

## 3.2. Principais Aspectos dos Algoritmos Genéticos

O primeiro passo de um Algoritmo Genético é a geração de uma população inicial de cromossomos que, normalmente, é formada por um conjunto aleatório de cromossomos que representam possíveis soluções do problema a ser resolvido. Durante o processo evolutivo, esta população é avaliada e cada cromossomo recebe um fitness, refletindo a qualidade da solução que ele representa.

Basicamente o AG pode ser descrito pelo seguinte algoritmo:

### **Algoritmo Genético**

1. Gerar a **população inicial**.
2. **Avaliar** cada indivíduo da população.
3. Enquanto **critério de parada** não for satisfeito faça
  - 3.1 Selecionar os indivíduos **mais aptos**.
  - 3.2 **Criar** novos indivíduos aplicando os operadores **crossover** e **mutação**.
  - 3.3 Armazenar os novos indivíduos em uma **nova população**.
  - 3.4 Avaliar cada cromossomo da nova população.

Em geral, os cromossomos mais aptos são selecionados e os menos aptos são descartados (Darwinismo). Os membros selecionados

podem sofrer modificações em suas características fundamentais através dos operadores de cruzamento e mutação, gerando descendentes para a próxima geração. Este processo é repetido até que uma solução satisfatória seja encontrada. O algoritmo acima ilustra este procedimento e, a seguir, estão descritos cada passo do algoritmo.

### 3.2.1. A Função objetivo (fitness)

Segundo Miranda [Miranda, 2001], a função objetivo de um problema de otimização é construída a partir dos parâmetros envolvidos no problema. Ela fornece uma medida de proximidade da solução em relação a um conjunto de parâmetros. Esta medida (que pode ser um percentual) é chamada de fitness.

Os parâmetros da função objetivo podem ser contínuos ou discretos. Otimização com parâmetros contínuos possui um número infinito de possíveis soluções, resultante de certa combinação de parâmetros.

No caso de um problema de corte de chapas, a função fitness deve envolver valores referentes à dimensão da chapa, a área desperdiçada com o corte, a demanda e o número de peças necessárias.

No presente trabalho, como o fitness diz respeito a um número muito grande de restrições, pode-se admitir um pré-cálculo desse valor envolvendo apenas algumas dessas restrições, que recebe o nome de subfitness; o fitness, então, é formado pela combinação de todos os subfitness registrados no problema.

### 3.2.2. Restrições

Pode-se definir como restrições um conjunto de equações ou inequações, que restringem o processo de tomada de decisão, como por exemplo, uma quantidade máxima disponível de algum recurso ou, ainda, a obrigatoriedade de uso mínimo de algum material ou recurso em um determinado processo.

Embora os algoritmos genéticos tenham sido inicialmente aplicados com sucesso em problemas de otimização sem restrições, foi apenas uma questão de tempo para surgirem as primeiras utilizações da técnica na resolução de problemas com restrições.

Uma abordagem para lidar com restrições nos AG's é simplesmente atribuir aptidão zero aos cromossomos inactiváveis.

Na maioria das aplicações são empregadas funções de penalização, de forma a “embutirem” as restrições na função objetivo, técnica esta, também, utilizada por uma parcela de métodos matemáticos de otimização.

### 3.2.3. Critérios de parada

Por se tratar de um algoritmo heurístico, e não exato, os Algoritmos Genéticos requerem um critério de parada para o mesmo.

Segundo BARCELOS [Barcelos, 2000], alguns dos possíveis critérios de parada para os AG's, são:

1. Quando o AG atingir certo número de gerações;
2. Chegada a um determinado valor da função objetivo definida a priori;
3. Convergência, isto é, quando não ocorrer melhoramento significativo no cromossomo de maior aptidão por um dado número de gerações.

Outras alternativas podem ser adotadas, como por exemplo, considerando que o processo evolutivo converge se 90% da população têm o mesmo valor para este cromossomo. Se entre 90% e 95% dos cromossomos convergirem, supões-se que, o AG convergiu para um ótimo local.

### 3.2.4. Representação e codificação

A escolha das variáveis que representarão matematicamente o problema a ser otimizado é de fundamental importância, pois terá impacto direto no desempenho da busca pela solução do mesmo. Portanto deve ser o mais simples possível sem, no entanto, perder as características de representação que o problema exige.

Segundo Holland [HOLLAND, 1975], existem inúmeras formas de representação das variáveis, tais como: binária, números inteiros e números reais. A maioria dos trabalhos desenvolvidos utiliza a codificação binária, em que cada cromossomo é um vetor, composto por zeros e uns, com cada bit representando um gene, o que não será o caso deste estudo, pois cada

gene, indicado por um numero inteiro, indicará uma quantidade de peças a serem cortadas.

A Figura 3.1 ilustra uma estrutura de cromossomos com seus subfitness (SF) e fitness(F) [Silva, 2005].

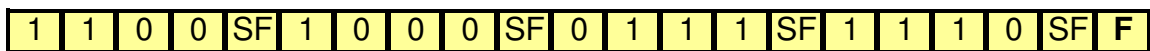


Figura 3.1 Cromossomo

Todos os demais conceitos serão exemplificados em um cromossomo desta natureza.

Problemas de otimização com variáveis reais, podem ser trabalhados em uma representação binária, mas para a verificação da aptidão (calculado do fitness) será necessária a conversão desses valores em pontos flutuantes, e as constantes conversões entre um tipo de representação e outro, podem tornar o uso do sistema binário desfavorável.

### 3.2.5. Geração da população inicial

A geração da população inicial de indivíduos, ou cromossomos, pode ser feita de forma aleatória, mas em alguns casos, principalmente quando se sabe que certo grupo de indivíduos atende determinados requisitos do problema, faz-se uma seleção heurística para que a população inicial carregue “bons indivíduos” sob o ponto de vista do que se espera da solução do problema.

Dado um número  $p$  de indivíduos, podendo ser gerados aleatoriamente ou não, a população inicial que pode ser representada como na Figura 3.2:

1-	0	1	0	0	SF	1	1	0	0	SF	0	1	0	1	SF	1	1	1	0	SF	F
2-	0	1	0	0	SF	1	1	0	0	SF	0	1	0	1	SF	1	1	1	0	SF	F
3-	1	1	1	0	SF	0	1	0	1	SF	0	0	0	1	SF	0	1	1	1	SF	F
p-	1	0	0	1	SF	0	1	0	0	SF	0	1	0	1	SF	0	0	1	0	SF	F

Figura 3.2 População Inicial.

Se uma população inicial for gerada aleatoriamente, espera-se que todas as regiões de busca do espaço estejam representadas. Este problema pode ser minimizado gerando uma população inicial com uma distribuição uniforme. Outra alternativa é gerar a primeira metade da população aleatoriamente e a segunda metade a partir da primeira, invertendo os bits. Isso garante que cada posição da cadeia de bits tenha um representante na população com valores binários.

Pode ser interessante usar uma população inicial maior que a utilizada nas gerações subsequentes, visando melhorar a representação do espaço de busca.

Uma técnica denominada “seeding” pode ser útil em vários problemas práticos. Consiste em colocar, na população inicial, soluções encontradas por outros métodos de otimização. Isso garante que a solução gerada pelos AG’s seja, no mínimo, tão boa quanto a gerada por esses métodos.



### 3.2.6. Avaliação da população

Depois das modificações geradas pelas operações de crossover ou mutação, é realizada a avaliação de todos os cromossomos da geração em questão. Uma função denominada fitness, é utilizada para esse fim, esta identifica a “qualidade” de cada indivíduo da população, e deve estar ligada à função objetivo cujo extremo deseja-se conhecer.

### 3.2.7. Processo de seleção

Segundo Holland, a seleção dos indivíduos da população baseia-se no princípio da “sobrevivência dos melhores indivíduos”, desta forma, os cromossomos melhor avaliados (segundo a função fitness) são copiados para um novo conjunto que formará a próxima geração, denominada população temporária. Em contrapartida, os indivíduos com baixa aptidão serão descartados da população. [HOLLAND, 1975].

#### 3.2.7.1. Processo de seleção por torneio

Neste método são escolhidos aleatoriamente (com probabilidades iguais)  $n$  cromossomos da população, e, o cromossomo com maior aptidão é

selecionado para a população intermediária. O processo repete-se até preencher a população intermediária. Utiliza-se, geralmente, o valor  $n = 2$ .

A seleção por torneio não precisa de escalonamento da aptidão e nem de ordenamento. Em uma ou outra versão, a seleção por torneio utiliza probabilidades diferenciadas. Se o torneio envolve dois cromossomos, o primeiro ganha o torneio com a probabilidade  $q$  (onde  $0,5 < q < 1$ ), e o segundo, com a probabilidade  $(1-q)$ . Para um torneio entre  $n$  cromossomos, o primeiro ganha o torneio com probabilidade  $q$  o segundo com probabilidade  $q(1-q)$ , o terceiro com  $q(1-q)^2$ , e assim por diante (vale notar que  $n = N$ , tal seleção é equivalente à seleção com ordenação exponencial).

Ao se aumentar o número  $n$  de cromossomos do torneio ou a probabilidade  $q$  do primeiro cromossomo vencer, aumenta-se a pressão de seleção, isto é, cromossomos com aptidão acima da média terão mais chance de serem selecionados.

### 3.2.7.2. Método da Roleta

Segundo Silva [Silva, 2005], um método largamente empregado para seleção dos indivíduos que sofrerão alteração genética para a formação de novos indivíduos é o método conhecido como “Método da Roleta”, em que cada indivíduo da população é representado de forma proporcional ao seu valor de aptidão. Assim, os indivíduos com alta aptidão recebem uma porção maior da roleta, enquanto os de baixa aptidão ocuparão uma porção

relativamente menor. Deste modo, a roleta é rodada tantas vezes quanto o número de indivíduos da população, escolhendo-se, assim, aqueles que darão origem à próxima população. Segue um exemplo na Figura 3.3. que mostra que indivíduos com maior aptidão, como é o caso do indivíduo 5, tem mais chance de ser selecionado (chance de 23,08%) e um indivíduo com baixa aptidão, como o indivíduo 1, possui menor chance de ser selecionado (chance de 17,31%).

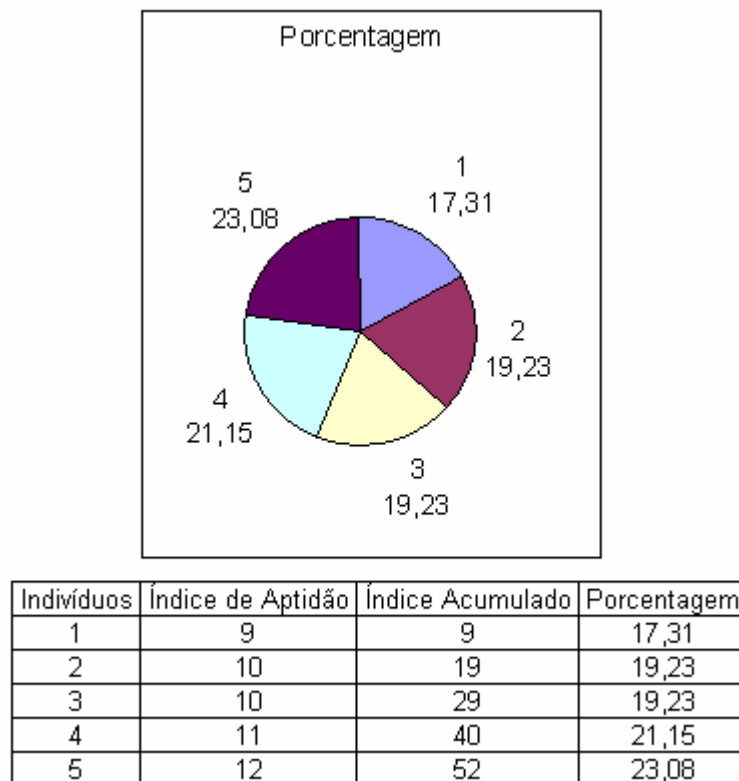


Figura 3.3 Método da Roleta.

Segundo Holland [Holland, 1975], o Método da Roleta tem a desvantagem de possuir alta variância, podendo levar a um grande número de cópias de um bom cromossomo, diminuindo a diversificação da população.

Uma outra idéia, entretanto, é abandonar a magnitude da aptidão de cada elemento e utilizar somente a posição de cada indivíduo na população. Deste modo, mantendo-se a população ordenada por valores decrescentes da aptidão, a probabilidade de seleção de um indivíduo para a etapa de recombinação cresce com o seu “ranking”, ou seja, o primeiro “ranking” tem maior probabilidade de seleção.

### 3.2.8. Reprodução ou cruzamento

Dependendo do sistema, geralmente, segundo Davis (DAVIS, 1991), o processo de seleção não introduz novos indivíduos na população, apenas os chamados progenitores, que servirão como pais para a nova geração, composta pelos filhos.

Na etapa de reprodução, o algoritmo tenta criar novas e melhores soluções (indivíduos mais aptos). A forma mais simples e natural de cruzamento é o chamado “cruzamento em um ponto”.

Como passo inicial, toda a população temporária, de tamanho  $n$ , é agrupada aleatoriamente por pares para gerar um conjunto de  $n/2$  progenitores potenciais.

O cruzamento é aprovado por um processo de decisão semelhante ao de um “lançamento de moeda”. Para isso, deve-se escolher uma probabilidade de cruzamento  $P_c$  igual para cada par. O lançamento de moeda corresponde a gerar, para cada par, um valor aleatório entre 0 e 1. Se

o valor gerado for inferior a  $P_c$ , o cruzamento é permitido, caso contrário, os progenitores são mantidos inalterados.

No processo de reprodução, ou cruzamento, deve-se aplicar um operador genético sobre os progenitores de forma a serem formatados novos indivíduos.

### 3.2.8.1. Operadores genéticos

Os operadores genéticos transformam uma população através de sucessivas gerações, de forma a obter um resultado que ao final do processo, seja satisfatório sob o ponto de vista do objetivo que se deseja alcançar. Através destes operadores, as características dos cromossomos são recombinadas e modificadas para a obtenção da diversidade da população.

#### 3.2.8.1.1. Crossover

Com o crossover de pelo menos dois progenitores, uma ou mais novas soluções são criadas a partir do intercâmbio de suas informações genéticas, em um ou mais pontos que também são selecionados aleatoriamente (OLIVEIRA, 1998). Este operador pode ser utilizado de várias maneiras. As mais empregadas são:

- Cruzamento de um ponto;

- Cruzamento multipontos;
- Cruzamento uniforme;

### 3.2.8.1.1.1. Crossover de um Ponto

Os operadores de crossover e de mutação são os principais mecanismos dos AG's para explorar regiões desconhecidas do espaço de busca. O operador crossover é aplicado a um par de cromossomos retirando da população intermediária, gerando dois cromossomos filhos. Cada um dos cromossomos pais tem sua cadeia de bits cortada em uma região aleatória, produzindo duas cabeças e duas caudas. As caudas são roçadas, gerando dois novos cromossomos. A figura 3.4. ilustra o comportamento deste operador. E, na seqüência, os filhos gerados por esta operação na Figura 3.5.



Figura 3.4. Crossover de um Ponto.

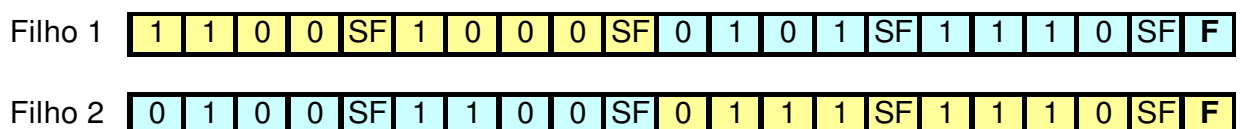


Figura 3.5. Filhos gerados pelos pais 1 e 2.

### 3.2.8.1.1.2. Cruzamento Multipontos

Segundo Cazangi, [Cazangi, 2002], o mecanismo é análogo ao operador anterior sendo que, ao invés de haver um ponto de corte, existem dois ou mais pontos escolhidos aleatoriamente. Na Figura 3.6, é mostrado como é feito o cruzamento de dois pontos, trocando os segmentos dos indivíduos em três partes.

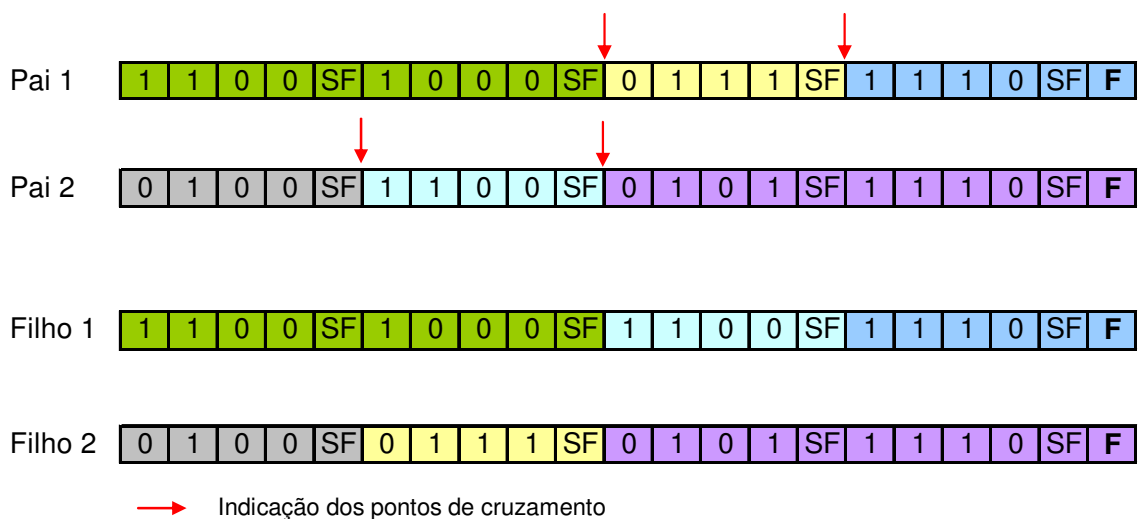


Figura 3.6 Cruzamento multipontos.

### 3.2.8.1.1.3. Cruzamento Uniforme

Neste tipo de cruzamento, cada gene descendente é criado copiando o gene correspondente de um dos pais, escolhido de acordo com uma espécie de gabarito de cruzamento gerado aleatoriamente [Cazangi, 2002]. Onde houver 1 no gabarito de cruzamento, o gene correspondente será copiado do primeiro pai e, onde houver 0 será copiado do segundo pai. O processo é repetido com os pais

trocados para produzir o segundo descendente. Um novo gabarito de cruzamento é criado para cada par de pais. Um exemplo pode ser visto na Figura 3.7:

Máscara de cruzamento	1	1	0	1		0	1	0	1		1	0	0	1		1	1	0	0		
Pai 1	1	1	0	0	SF	1	0	0	0	SF	0	1	1	1	SF	1	1	1	0	SF	F
Filho	1	1	0	0	SF	1	0	0	0	SF	0	1	0	1	SF	1	1	1	0	SF	F
Pai 2	0	1	0	0	SF	1	1	0	0	SF	0	1	0	1	SF	1	1	1	0	SF	F

Figura 3.7 Cruzamento Uniforme.

### 3.2.8.1.2. Mutação

Este operador é aplicado a uma taxa muito pequena da população, pois se trata de um operador genético secundário, porém ele é extremamente necessário para a introdução e manutenção da diversidade genética dessa população, pois ele trabalha para a introdução de novos elementos nessa população.

Após a operação de cruzamento, o operador de mutação é aplicado, com uma determinada probabilidade, em cada bit dos dois filhos. O operador inverte o operador de bits, de 1 para 0 ou de 0 para 1. A figura 3.8 representa um exemplo do processo de mutação. A mutação melhora a diversidade dos cromossomos na população, porém, destrói parte da informação contida, logo, deve ser utilizada uma taxa de mutação pequena



(normalmente entre 0,1% e 5%), mas, suficiente para assegurar a diversidade, veja Figura 3.6.

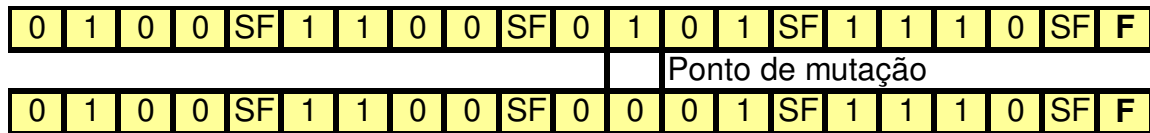


Figura 3.8 Ponto de mutação.

### 3.2.8.1.3. Elitismo

Nos AG's, quando se geram novas populações utilizando os operadores mostrados nos itens anteriores, como cruzamento e mutação, pode-se facilmente observar que, em muitos casos, o melhor indivíduo de uma geração desaparece na geração seguinte. Para melhorar a eficiência dos AG's é interessante transferir o melhor cromossomo de uma geração para outra sem alterações. Esta estratégia é denominada Elitismo, sendo muito comum nos AG's tradicionais.

O elitismo foi proposto em 1975 por De Jong (1993), em um dos trabalhos pioneiros sobre AG's, sendo que em cada geração, o indivíduo com menor (em problema de minimização) valor de fitness é copiado para a geração seguinte sem alterações, garantindo, desta forma, que sempre a melhor solução encontrada em qualquer uma das gerações será mantida até o final do processo.

## 4. Algoritmo Genético Aplicado a um Problema de Corte Bidimensional.

A seguir, será descrito um AG modelado ao problema de corte bidimensional. Serão listadas as restrições que foram consideradas, a abordagem dada ao problema, a descrição das técnicas usadas. Para o desenvolvimento dos algoritmos, foi usado o programa Matlab/2006. Para os testes de otimização foi utilizado um pacote de métodos quantitativos, o QM (Quantitative Methods), version 2 (build 20) copyright 1996-2000 Howard J. Weiss, for Windows/2000.

### 4.1. Descrição do Problema

Uma empresa precisa cortar chapas de aço para gerar as peças que serão usadas na manufatura de seus produtos. Dessa forma, o problema de corte bidimensional a ser resolvido consiste em: dado um número finito  $m$  de itens retangulares (denominados peças) de largura  $l_i$ , comprimento  $c_i$  e demanda  $d_i$ , a serem obtidos de uma chapa de dimensões  $L \times C$ , encontrar padrões de corte da chapa que atendam a demanda de itens solicitados com o menor desperdício possível.

O objetivo do presente estudo é encontrar padrões de corte que minimizem o desperdício de material (chapas) utilizando Algoritmo Genético em condições de canibalização. Outra característica do objeto em estudo é que se está tratando de um problema restrito, isto é, o número de peças

possui uma demanda que deve ser respeitada.

Matematicamente, tem-se:

$$\begin{array}{ll} \text{Min} \sum_{j=1}^n P_j x_j & \text{Minimizar o desperdício de material de} \\ & \text{uma chapa.} \\ \\ \text{(M)} \quad \text{S.a.} \sum_{j=1}^n a_{ij} x_j \geq d_i \quad i = 1, \dots, m & \text{Sujeito a: O número de peças do tipo } i \\ & \text{geradas por todos os padrões de corte } j \\ & \text{deve ser maior ou igual à demanda da} \\ & \text{peça } i. \\ \\ x_j \geq 0, \quad j = 1, \dots, n \text{ (inteiro)} & \end{array}$$

As variáveis de decisão são:

$x_j$  - número de vezes que a faixa  $j$  é utilizada na chapa;

$a_{ij}$  - número de peças do tipo  $i$  obtidas da faixa  $j$ ;

$P_j$  - perda ou desperdício associado ao uso da faixa  $j$ .

$d_i$  - demanda da peça do tipo  $i$ . A demanda  $d_i$  é gerada por algum problema no processo de produção que acarretou a perda de um lote de peça, portanto, não é um número demasiadamente grande, dessa forma uma chapa deve dar conta de toda a demanda.

$m$  - número total de peças;

$n$  - número de padrões (faixas) considerados;

Dessa forma, tem-se um padrão de corte da faixa  $j$  representado por um vetor cujo  $i$ -ésimo componente é  $a_{ij}$  representando o número de peças do tipo  $i$  que são cortadas com o padrão de corte da faixa  $j$ .

Na verdade tem-se um problema de otimização simples, que poderia ser resolvido por um método exato, mas o processo de geração dos padrões de corte das faixas pode se tornar um processo demorados, e aliado à canibalização que exige respostas rápidas, por isso optou-se pelo uso de um método heurístico.

Os algoritmos heurísticos, como o AG, servem exatamente para se resolver problemas de elevado nível de complexidade em tempo computacional razoável. Ao se pensar em um problema combinatório, uma opção seria analisar todas as combinações possíveis para conhecer a melhor. Se o problema possui um universo de dados pequeno, realmente esta é a maneira correta de se buscar a melhor solução, mas os problemas reais, normalmente, possuem um número de combinações muito extenso, o que torna inviável a análise de todas as combinações. As heurísticas procuram encontrar soluções próximas da otimalidade em um tempo computacional razoável.

#### 4.1.1. Hipóteses

- O tipo de corte abordado neste trabalho é do tipo guilhotina e as peças e as chapas são do tipo retangular, do que se pode afirmar que o corte será do tipo ortogonal, ou seja, paralelo um dos lados da chapa principal;
- As máquinas que farão o corte deverão ser do tipo guilhotina, em que a lâmina, quando faz o corte, percorre a chapa de uma extremidade a outra dividindo a chapa em duas partes;

- A medida que corresponde a espessura da lâmina na chapa não será observada neste trabalho;
- Neste modelo, propõe-se o corte não estagiado, pois não há restrições quanto ao número de estágios de corte;
- Tanto as peças quanto as chapas deverão ser representados por duas dimensões altura e largura e sua área o produto das duas dimensões;
- Assume-se, para este trabalho, que os valores correspondentes às dimensões terão números inteiros e serão dados em milímetros (mm);
- Durante a projeção das peças, estas devem estar sempre se tocando de forma parcial ou total, lado a lado, o que se denomina de corte normalizado [Christofides, 1995];
- As peças podem ter dimensões homogêneas, ou seja, a mesma altura e a mesma largura;
- As peças podem girar num ângulo de  $90^\circ$ ;
- As peças não podem ser sobrepostas;

- Admite-se que a área considerada da chapa é representada pela área total ocupada pelas peças e mais a área de desperdício interno.

Neste trabalho, busca-se a minimização da função objetivo que considera a perda total de material no atendimento da demanda na situação de canibalização.

#### 4.1.2. Geração da População Inicial.

Para descrever os procedimentos que demonstram de que forma foi realizada a implementação do AG usar-se-á a construção de um padrão de corte com apenas duas peças.

Considere-se a Tabela 4.1 que representa uma pequena amostra de dados, em que a chapa que será usada tem dimensões  $L=1130 \text{ mm}$  (largura – sentido horizontal) por  $C=2736 \text{ mm}$  (altura – sentido vertical). Nesta chapa pretende-se cortar peças com as dimensões  $l=697 \text{ mm}$  (largura) por  $c=322 \text{ mm}$  (comprimento), e  $l=298 \text{ mm}$  (largura) por  $c=122 \text{ mm}$  (comprimento), com demanda de 8 e 5 peças respectivamente. Toda a demanda poderá ser atendida por uma única chapa.

Dimensão da chapa		Placa		Consumo diário
L	C	l	c	d
1130	2736	697	322	8
1130	2736	298	122	5
1130	2736	450	145	5
1130	2736	245	145	5
1130	2736	676	157	8

Tabela 4.1 Dimensões e demandas.

A partir desse conjunto de dados, desenvolve-se um padrão de corte em forma de faixas, ou seja, dada a largura da chapa ( $L$ ) definem-se quantas peças, no máximo, podem-se encaixar nessa dimensão. A Figura 4.1. mostra o padrão de corte de uma faixa com três peças de 298 mm por 122 mm e uma peça de 697 mm por 322 mm, que estão em vermelho, mais a área desperdiçada, que esta em amarelo.

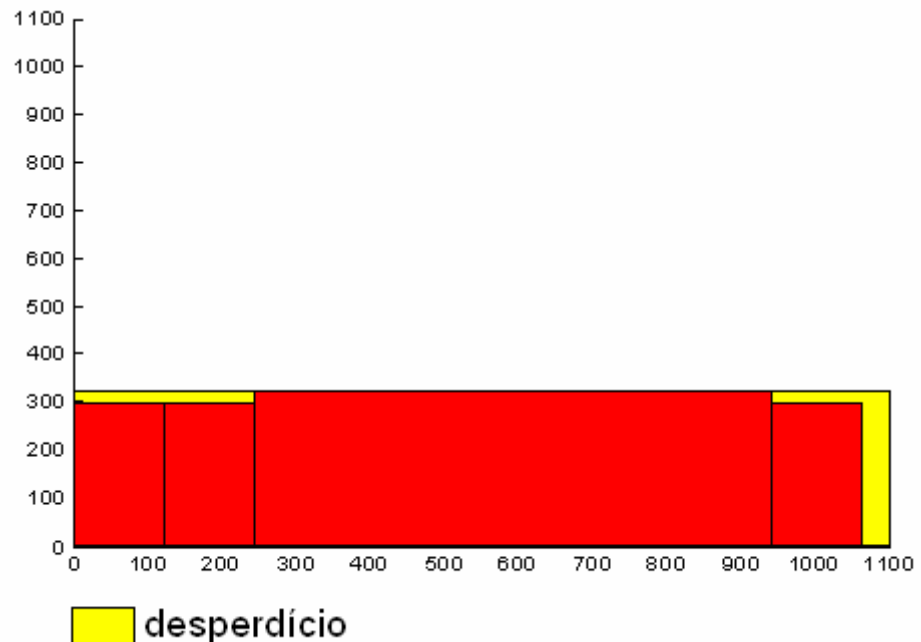


Figura 4.1. Exemplo de Plano de Corte de uma Faixa.

Dessa maneira, tem-se o padrão de corte de uma faixa dessa chapa, mas, ao se combinarem várias faixas, tem-se um padrão de corte para uma chapa inteira (Figura 4.2). Nessa combinação, observa-se o cuidado com o comprimento ( $C$ ) da chapa para que as dimensões das faixas selecionadas sejam compatíveis à chapa usada. Cada faixa pode gerar um desperdício, que é a perda de material, mas a combinação das

faixas, quando não atinge todo o comprimento da chapa, diferentemente da perda  $P_j$  considerada no modelo (M) da página 43 gera uma sobra que pode ser reutilizada ou vendida, pois se trata de uma tira inteira de material.

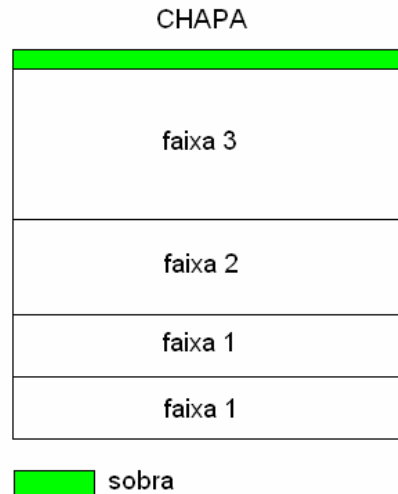


Figura 4.2. Várias faixas formando um padrão de corte da chapa.

#### 4.1.2.1. Dados de Entrada do Algoritmo Genético:

##### Construção das faixas.

Para a utilização do AG precisa-se de uma população inicial de cromossomos que, neste caso, será formada por um conjunto de faixas combinadas com seus respectivos valores de subfitness (SB) e fitness (F). A Figura 4.3 exemplifica a estrutura do padrão de corte formado pela combinação das faixas em forma de cromossomo, utilizados no AG proposto.



Figura 4.3 Padrão de corte da chapa em forma de cromossomo.



A construção dessas faixas, que formarão os chamados genes dos cromossomos, será feita por um algoritmo denominado *FAIXAS* que foi desenvolvido nesse trabalho especialmente para esse fim. Esse se baseia em duas heurísticas, o *First Fit Decreasing* e o *First Fit Decreasing Height*, que incorporam a idéia de ordenação de peças e empacotamento [Teodoro, 2003], detalhadas no capítulo 2 deste trabalho.

#### 4.1.2.2. Descrição do Algoritmo de Construção das Faixas.

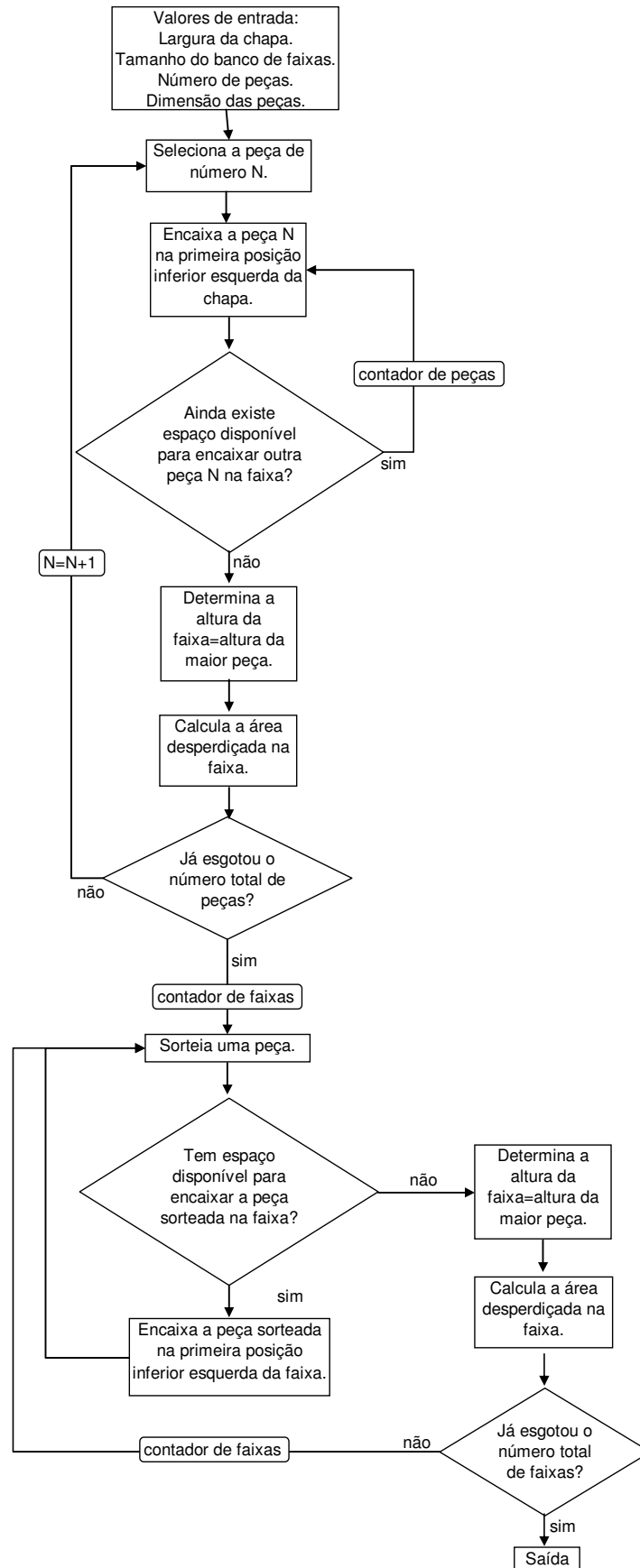
Para a definição da dimensão de cada gene, foram levadas em consideração as características dos parâmetros de entrada. A partir dos dados: dimensão da chapa, tamanho do banco de faixas, número de peças e dimensão das peças a serem cortadas, é gerado o padrão de corte das faixas e a uma população inicial de cromossomos para as simulações.

O algoritmo gera faixas de duas formas diferentes:

- Sorteia uma peça e completa a faixa com a mesma peça sorteada;
- Completa a faixa inteira por sorteio;

A saída do algoritmo nos dá uma tabela, onde se tem esses padrões de corte das faixas, com o número de peças cortadas em cada faixa, a altura da faixa e o desperdício que aquele padrão gera.

Segue, abaixo, o Fluxograma 4.1 do algoritmo *FAIXAS*, o programa completo está no Anexo 1.



Fluxograma 4.1 Algoritmo FAIXAS.

Na tabela abaixo seguem os quatorze padrões de corte das faixas gerados pelo Algoritmo Faixas, com os dados das duas primeiras peças da Tabela 4.1. (Tabela 4.2).

Quantidade de peças		altura	perda (mm <sup>2</sup> )
Peça 1	Peça 2	da faixa	(desperdício)
0	3	122	28792
3	0	697	114308
0	9	298	9536
1	3	697	454108
1	3	322	30358
2	0	697	338742
1	5	697	381396
2	3	697	229674
1	3	697	454108
2	0	697	338742
1	3	322	30358
1	3	322	30358
1	3	322	30358

Tabela 4.2 Saída do Algoritmo Faixas.

Para cada faixa, o algoritmo gera uma representação geométrica para melhor visualização da faixa sugerida. Para a terceira linha da tabela de saída, se têm os valores especificados na Tabela 4.3.:

0	9	298	9536
---	---	-----	------

Tabela 4.3. Terceira faixa da tabela 4.2

Que representam na prática, zero peças de dimensão 697 mm por 322 mm, e nove peças de dimensão 122 mm por 298 mm, a faixa terá altura de 298 mm e o desperdício gerado por esta padrão de corte é de 9536mm<sup>2</sup> ou 0,02% da área total da faixa. A representação geométrica segue na Figura 4.4.

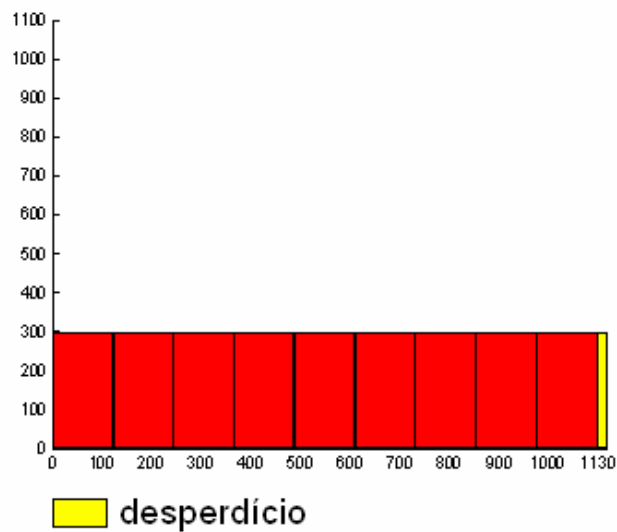


Figura 4.4. Representação Geométrica da Terceira Faixa.

### 4.1.2.3. Descrição do Algoritmo de Montagem do Cromossomo.

Os cromossomos são formados por genes, que nesse trabalho correspondem às faixas construídas anteriormente e detalhados na tabela 4.2. Para a construção desses cromossomos, que nesse trabalho representam os padrões de corte da chapa, um algoritmo chamado *TESTE* foi desenvolvido. O nome faz referência a uma das operações que o algoritmo realiza que é a de testar a factibilidade desse cromossomo, ou seja, à medida que os cromossomos são criados, duas saídas são avaliadas:

- O desperdício de material que o uso deste padrão de corte da chapa produz, calculada em milímetros quadrados;
- Se este padrão atende ao menos entre 80% e 120% da demanda no processo de canibalização, valor considerado como mínimo

para que esse cromossomo faça parte da população inicial (atendendo 80% da demanda o cromossomo ainda é inactivável, mas essa restrição é usada somente para criar uma população inicial mais diversificada);

As duas saídas citadas acima são usadas no cálculo do fitness do cromossomo, que avalia e define se esse cromossomo é bom ou não. Uma vez gerados os cromossomos da população inicial, ou gerado um novo cromossomo a partir de uma operação genética, deve-se calcular o fitness de cada um deles.

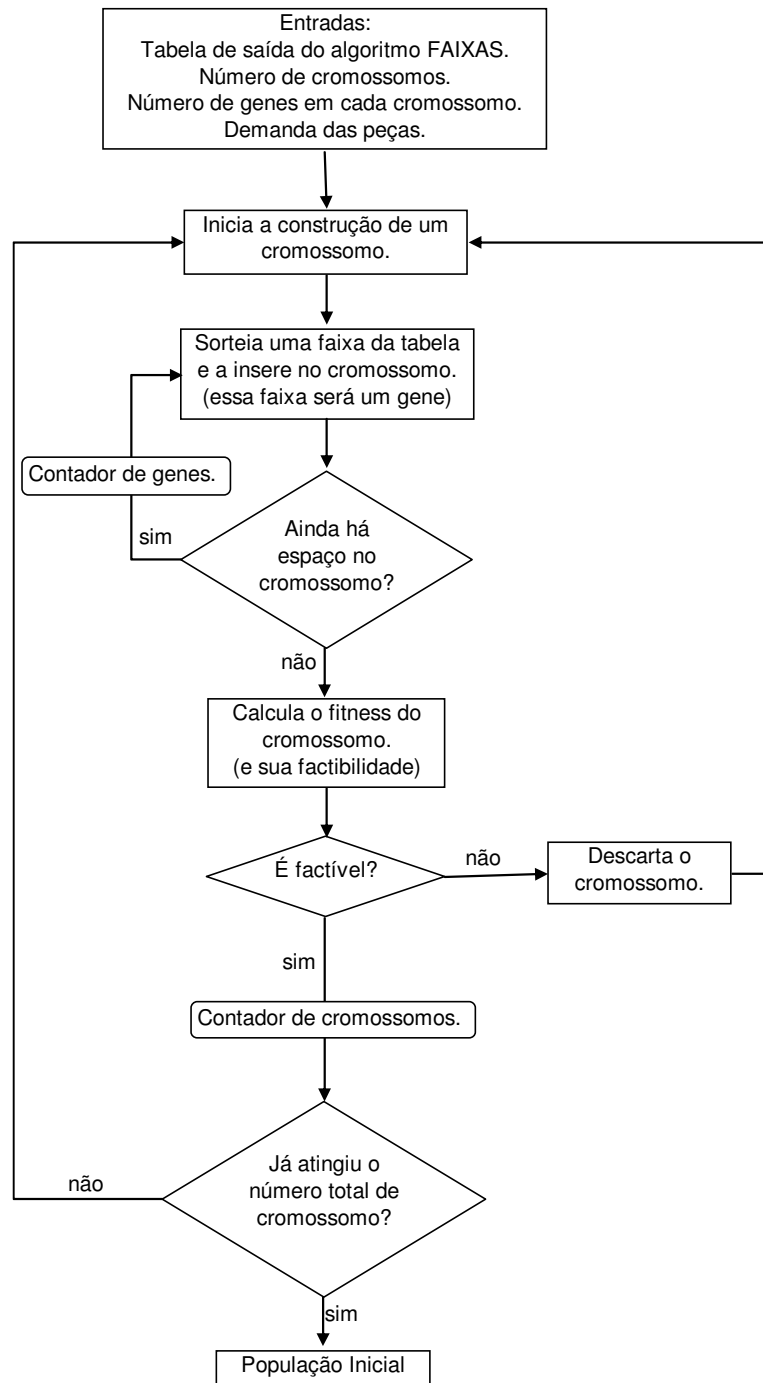
Como cada cromossomo corresponde a um padrão de corte da chapa que é composto de padrões de corte de faixas, esse deve ter o menor desperdício possível, atendendo, ao menos em parte, a demanda do problema. Considerando-se estes fatos, definiu-se como melhor cromossomo aquele que possui atendimento da demanda (para este problema, em particular, podem sobrar, mas não pode faltar peças) e que também possua a menor perda.

O valor do sub-fitness é o desperdício de material de cada gene (ou faixa), calculado em milímetros quadrados, desconsiderando-se para esse cálculo a demanda, pois, para atenderem-se as quantidades de peças precisa-se do cromossomo inteiro. Precisa-se, também, do fitness, que é calculado somente depois da obtenção de todo o cromossomo, ou seja, depois de selecionadas as faixas que compõem esse padrão de corte chamado cromossomo, levando em conta todos os sub-fitness já calculados. Define-se o fitness pela seguinte fórmula:

$$Fitness = \frac{\textit{porcentagem de perda de material na chapa}}{\textit{porcentagem de atendimento da demanda com uma chapa}}$$

No fitness é considerado o percentual de atendimento da demanda e a área total desperdiçada.

Segue, abaixo, o Fluxograma 4.2 do algoritmo *TESTE*, o programa completo está no Anexo 2.



Fluxograma 4.2 Algoritmo *TESTE*.

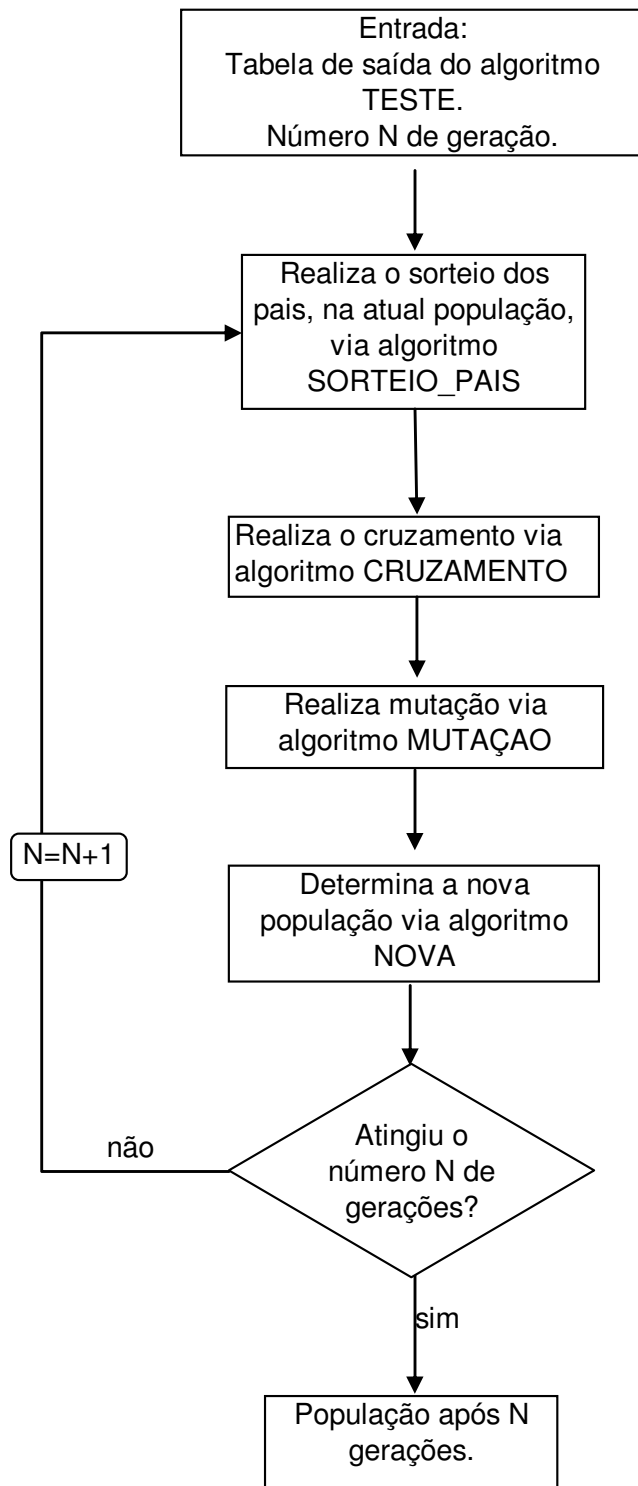
A saída desse algoritmo é uma família de cromossomos, todos testados e factíveis, apresentados em forma da Tabela 4.4.

Peças do tipo 1	Peças do tipo 2	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Altura da faixa	Perda de material na faixa	Perda de material no cromossomo	Fitness	Percentual de perda no cromossomo
1	3	322	30358	3	0	697	114308	1	3	322	30358	1	3	322	30358	205382	0,0810	10,93%
0	3	122	28792	0	3	122	28792	1	3	322	30358	3	0	697	114308	202250	0,1575	14,17%
2	3	697	229674	1	5	697	381396	2	3	697	229674	2	3	697	229674	1070418	0,1387	33,98%
1	3	322	30358	1	3	697	454108	1	3	697	454108	3	0	697	114308	1052882	0,2860	38,61%
1	3	322	30358	1	5	697	381396	1	0	322	139426	1	3	322	30358	581538	0,2813	30,95%
1	3	322	30358	2	3	697	229674	2	0	697	338742	1	3	322	30358	629132	0,2024	27,32%
2	3	697	229674	1	5	697	381396	1	3	697	454108	1	3	322	30358	1095536	0,2296	40,18%
2	3	697	229674	1	3	322	30358	1	3	697	454108	0	9	298	9536	723676	0,1767	31,80%
2	0	697	338742	1	3	322	30358	2	0	697	338742	1	3	697	454108	1161950	0,4735	42,61%
0	9	298	9536	1	0	322	139426	1	3	697	454108	2	3	697	229674	832744	0,2439	36,59%

Tabela 4.4 Cromossomos População Inicial.

#### 4.1.2.4. Descrição do Algoritmo Genético.

A partir da construção da população inicial, começa-se a aplicação dos operadores genéticos sobre essa população. O Fluxograma 4.3 (programa completo está no Anexo 3) descreve a seqüência de operações realizada pelo AG e cada um desses operadores é explicado na secção seguinte. O critério de parada adotado para o AG é o limite máximo de gerações do algoritmo e apresentará, também em forma de tabela, os cromossomos de melhor fitness como resultado do problema.

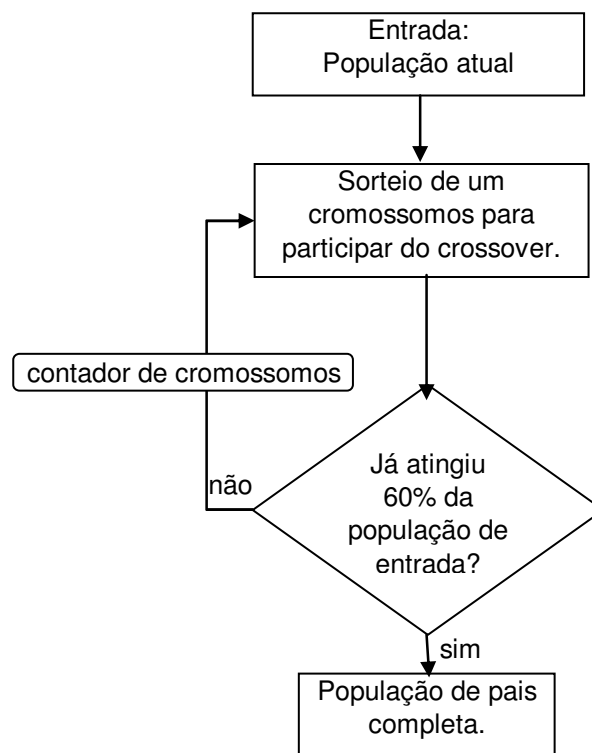


Fluxograma 4.3 Operadores Genéticos.



## CROSSOVER

Para aplicação do operador crossover, faz-se primeiramente um sorteio, usando permutação randômica para a escolha dos pais que participarão desse cruzamento. Considera-se, aqui, que a população de pais é 60% do tamanho da população inicial. O Fluxograma 4.4 explica o algoritmo denominado SORTEIO\_PAIS em que é feito esse sorteio (o programa completo está no Anexo 4).



4.4 Fluxograma Explicativo do Algoritmo SORTEIO\_PAIS.

Para realização do crossover, os cromossomos são agrupados em duplas e determina-se por sorteio um ponto de cruzamento entre os pais

selecionados, realiza-se então, a troca de informações genéticas (genes). O esquema nas Figuras 4.5 e 4.6 ilustram o processo de crossover, considerando-se que o ponto de cruzamento sorteado fica entre o segundo gene e o terceiro gene.

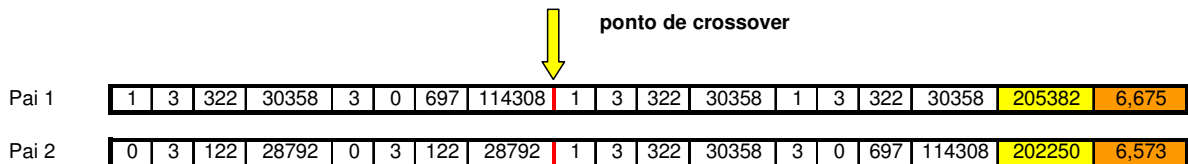


Figura 4.5. Ponto de crossover.

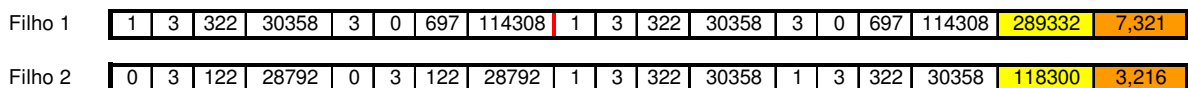
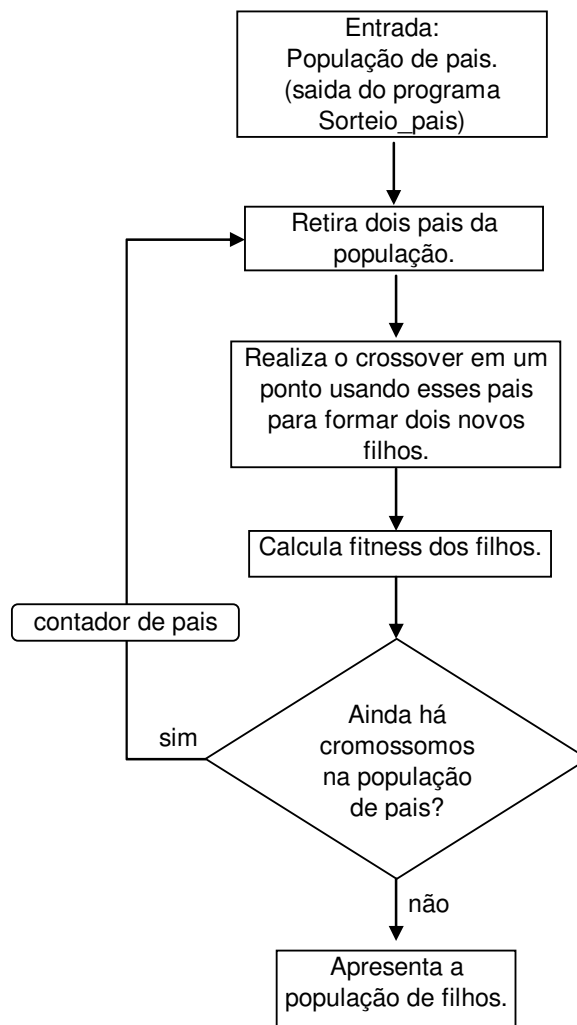


Figura 4.6. Cromossomos mutados e com novo fitness.

A partir do ponto de crossover o cromossomo 1 (pai 1) é reescrito usando-se a informação contida no cromossomo 2 (pai 2), o mesmo acontece com o cromossomo 2 que, a partir do ponto de crossover, é reescrito usando-se a informação contida no cromossomo 1. Como resultado desse cruzamento, obtêm-se dois filhos, cada um com seu fitness recalculado, e se o cromossomo filho for infactível, ele é descartado automaticamente (via algoritmo). Todo esse processo é realizado pelo algoritmo CRUZAMENTO, descrito no Fluxograma 4.5 (o programa completo está no Anexo 5).

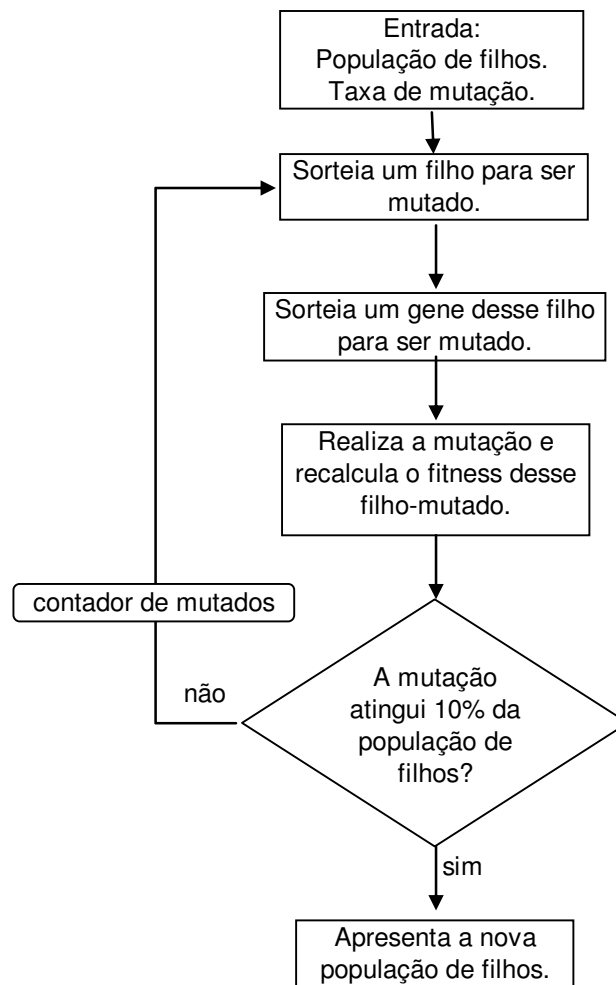


4.5 Fluxograma Explicativo do Algoritmo CRUZAMENTO.

A saída desse algoritmo é uma população de filhos do mesmo tamanho da população de pais, pois cada dois pais geram dois novos filhos, e o algoritmo só conclui uma geração quando esta estiver do tamanho da geração anterior, logo, mesmo que alguns filhos sejam descartados, outros serão calculados para ocuparem estes lugares. Nessa população de filhos ocorrerá o processo de mutação.

## **MUTAÇÃO**

Por se ter em vista que o operador de mutação é necessário para a introdução e manutenção da diversidade genética da população, foram mutados 10% da população de filhos. A mutação pode ocorrer dentro de um cromossomo em qualquer dos seus genes. Nesses pontos, o gene selecionado é trocado por outro gene escolhido aleatoriamente, diferente do atual, escolhido dentro da população atual por sorteio. Caso a mutação gere uma infactibilidade, ou seja, se a troca for por um gene que não atenda às especificações do algoritmo FAIXAS, escolhe-se outro gene para a mutação. Depois da aplicação desse operador, o fitness desse cromossomo é recalculado. Veja o procedimento que descreve o algoritmo MUTAÇÃO no Fluxograma 4.6 (o programa completo está no Anexo 6).

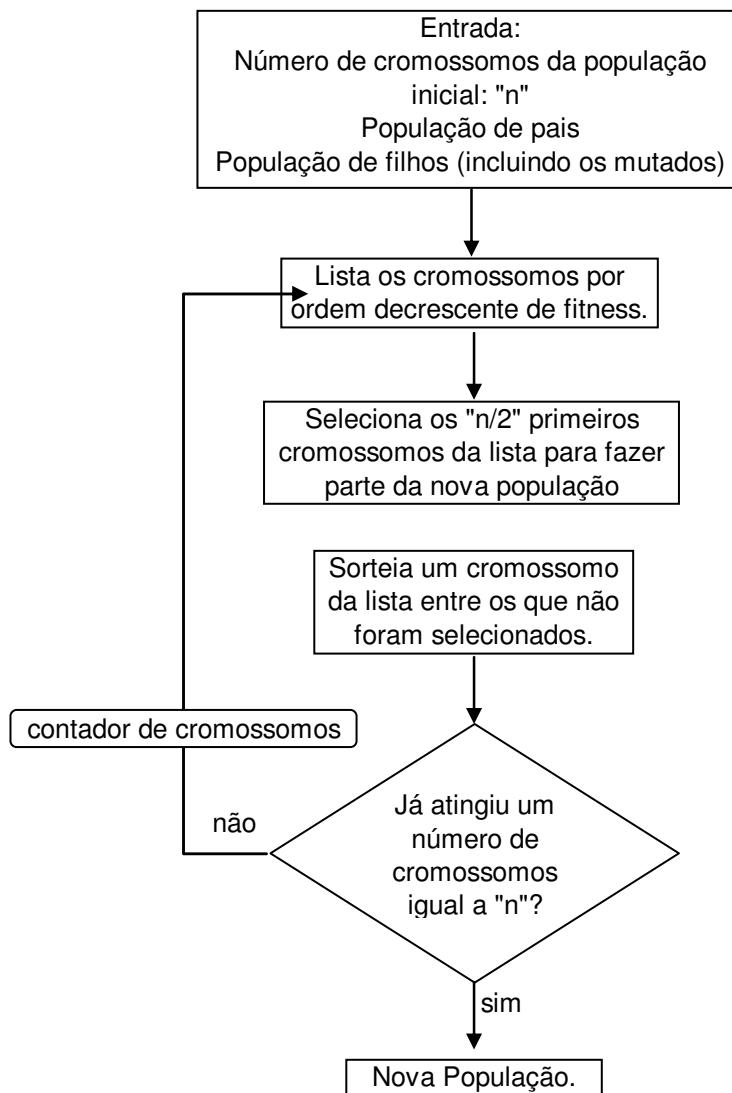


4.6 Fluxograma Explicativo do Algoritmo MUTAÇÃO.

### **NOVA POPULAÇÃO**

Depois de aplicados os operadores genéticos para a produção de filhos, uma nova população toma o lugar da população anterior. Os indivíduos dessa nova população são determinados da seguinte forma: primeiramente, são listados todos os pais e filhos por ordem decrescente de seus valores de fitness. Quanto menor o valor do fitness, melhor é

aquele padrão de corte, pois cumpre a demanda e menor é sua perda. Observa-se que mesmo quando a demanda seja ultrapassada as peças cortadas a mais não são consideradas desperdício. Como essa nova população terá o mesmo número de indivíduos que a população inicial, 50% dela será preenchida pelos primeiros cromossomos da lista, que serão evidentemente os mais adaptados, pois possuem melhores fitness; os outros 50% serão preenchidos por sorteio realizado entre os cromossomos não selecionados desta mesma lista. O algoritmo que faz esta seleção foi chamado de NOVA e está descrito no Fluxograma 4.7, que segue (o programa completo está no Anexo 7).



4.7 Fluxograma explicativo do algoritmo NOVA.

Com essa seqüência de algoritmos, espera-se chegar a uma população com valores de fitness menores a cada geração, de modo que se chegue cada vez mais perto da solução do problema. O critério de parada para o AG é o número de gerações que se deseja calcular, o que não significa chegar na melhor solução, mas que apresenta uma melhora nos

valores de fitness a cada nova geração, e, na solução (saída do algoritmo) apresentar-se-ão os melhores indivíduos como resultado.

A Tabela 4.5 mostra os cromossomos da vigésima geração do problema, inicialmente abordado neste capítulo. Comparando-se com a população inicial, apresentada na Tabela 4.4, observa-se uma grande melhoria nos valores dos fitness.

Peças do tipo		Altura da faixa	Perda de material na faixa	Peças do tipo		Altura da faixa	Perda de material na faixa	Peças do tipo		Altura da faixa	Perda de material na faixa	Peças do tipo		Altura da faixa	Perda de material na faixa	Perda de material no cromossomo	Fitness	Percentual de perda no cromossomo
1	2			1	2			1	2			1	2					
1	3	322	30358	3	0	697	114308	1	3	322	30358	0	9	298	9536	184560	0.0531	9,97%
1	3	322	30358	3	0	697	114308	1	3	322	30358	0	9	298	9536	184560	0.0531	9,97%
1	3	322	30358	3	0	697	114308	1	3	322	30358	0	9	298	9536	184560	0.0531	9,97%
1	3	322	30358	3	0	697	114308	1	3	322	30358	0	9	298	9536	184560	0.0531	9,97%
1	3	322	30358	3	0	697	114308	1	5	697	381396	1	3	697	454108	980170	0.2179	35,95%
1	3	322	30358	3	0	697	114308	2	0	697	338742	2	3	697	229674	713082	0.2179	26,15%
1	3	322	30358	3	0	697	114308	1	3	322	30358	0	9	298	9536	184560	0.0531	9,97%
1	3	322	30358	3	0	697	114308	1	3	322	30358	1	3	322	30358	205382	0.0810	10,93%
1	3	322	30358	3	0	697	114308	1	3	322	30358	0	9	298	9536	184560	0.0531	9,97%

Tabela 4.5 Cromossomos da vigésima geração.

Ao se analisar os resultados da tabela, percebe-se que os melhores padrões de corte são os de menor fitness, pois atendem a demanda, de forma que não falte nenhuma peça, e tem uma perda pequena. Dessa forma, os melhores padrões de corte são os cromossomos 1 e 2. O primeiro pede uma tiragem de 5 peças do tipo 1, 15 peças do tipo 2, atendendo a demanda em 185%, e com perda de 9,97% do material utilizado. O segundo pede uma tiragem de 6 peças do tipo 1, 9 peças do tipo 2, atendendo a demanda em 116%, e com perda de 10,93%. Cabe a Indústria decidir sobre a melhor opção, entre os fitness alcançados pelo AG, sendo que não é problema uma tiragem bem maior que o pedido.



## 4.2. Resultados.

Observa-se a seguir alguns testes e resultados aplicando-se as mesmas operações genéticas, levando-se em consideração as seguintes variações:

- Variação no número de iterações do AG (variação no número de gerações);
- Variação na demanda e no número de peças;
- As respectivas comparações com os resultados do mesmo problema resolvido por programação linear inteira com o auxílio do programa QM.

### **Primeiro Teste**

No primeiro teste utilizou-se a Tabela 4.10, e nesta as duas primeiras peças com suas respectivas demandas. Os testes consideram um banco de faixa de trinta dados (ou genes), e o número de gerações igual a vinte, a partir de uma população inicial.

Dimensão da chapa mãe		Blank		Consumo diário
1100	3468	122	298	11
1100	3468	145	450	14
1100	3468	145	245	15
1100	3468	61	480	13
1100	3468	696	460	10
1100	3468	714	381	15
1100	3468	344	489	14

Tabela 4.10 Dimensões e demanda.

Foi observado que para o primeiro teste, após 20 gerações houve convergência para o ótimo local, mas se recomenda para problemas maiores, um número maior de gerações. Evidentemente almeja-se chegar a um ótimo global, mas se aceita um ótimo local devido a urgência de decisão que deve ser tomada no processo.

Aplicando-se o algoritmo FAIXAS, para a construção dos genes, obtêm-se a Tabela 4.11. Esta tabela é apenas um exemplo de como acontece a saída dos dados, pois para cada vez que rodamos o programa AG, um novo banco de genes é construído.

Quantidade de peças		altura da faixa	Perda
Peça 1	Peça 2		desperdício
9	0	298	596
0	7	450	38250
3	0	122	25132
0	2	145	29000
2	1	145	21538
1	3	450	262894
3	2	450	255432
1	3	450	262894
0	3	450	299250
1	4	450	197644
4	1	450	284326
1	2	298	160944
3	1	298	153482
4	0	298	182376
2	3	450	226538
4	4	450	88576
3	1	298	153482
1	4	450	197644
1	3	450	262894
1	3	450	262894
1	2	298	160944
2	2	450	291788
1	2	298	160944
3	1	298	153482
4	3	450	153826
3	1	298	153482
1	5	450	132394
2	1	145	21538
4	2	450	219076
2	1	145	21538

Tabela 4.11 Faixas.

Para cada banco de genes é construída uma população inicial com quatro genes que é então submetida às operações genéticas para certo número de gerações. A população inicial associada à Tabela 4.11 está representada na Tabela 4.12.

Peças do tipo 1	Peças do tipo 2	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Altura da faixa	Perda de material na faixa	Perda de material no cromossomo	Fitness	Percentual de perda no cromossomo
3	0	122	25132	3	1	298	153482	1	3	450	262894	1	2	298	160944	602452	1,9049	46,89%
3	1	298	153482	1	3	450	262894	4	4	450	88576	3	1	298	153482	658434	0,7881	40,01%
1	3	450	262894	3	0	122	25132	1	2	298	160944	1	3	450	262894	711864	1,9917	49,03%
0	7	450	38250	4	3	450	153826	2	2	450	291788	1	4	450	197644	681508	0,5993	34,42%
1	4	450	197644	9	0	298	596	4	0	298	182376	4	4	450	88576	469192	0,3861	28,51%
1	3	450	262894	1	2	298	160944	9	0	298	596	0	7	450	38250	462684	0,4154	28,12%
1	4	450	197644	1	4	450	197644	2	1	145	21538	4	2	450	219076	635902	0,8569	38,67%
1	2	298	160944	1	2	298	160944	4	0	298	182376	1	3	450	262894	767158	2,0651	51,89%
1	2	298	160944	2	1	145	21538	3	1	298	153482	0	3	450	299250	635214	2,2511	48,49%
1	3	450	262894	3	1	298	153482	4	4	450	88576	3	1	298	153482	658434	0,7881	40,01%

Tabela 4.12 População inicial.

Na vigésima geração observa-se uma melhoria muito boa dos fitness desses cromossomos. Ao analisar-se cada cromossomo da Tabela 4.13 individualmente, nota-se que diminui a perda de material e a demanda é atendida. O melhor cromossomo (considerando-se o fitness) repete-se diversas vezes, pois se preservam os melhores fitness e se descartam os piores.

Peças do tipo 1	Peças do tipo 2	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Altura da faixa	Perda de material na faixa	Perda de material no cromossomo	Fitness	Percentual de perda no cromossomo
0	7	450	38250	9	0	145	596	9	0	298	596	0	7	450	38250	77692	0,0789	5,26%
0	7	450	38250	9	0	145	596	9	0	298	596	0	7	450	38250	77692	0,0789	5,26%
0	7	450	38250	9	0	145	596	9	0	298	596	0	7	450	38250	77692	0,0789	5,26%
0	7	450	38250	9	0	145	596	9	0	298	596	0	7	450	38250	77692	0,0789	5,26%
0	7	450	38250	9	0	145	596	9	0	298	596	0	7	450	38250	77692	0,0789	5,26%
0	7	450	38250	9	0	145	596	9	0	298	596	0	7	450	38250	77692	0,0789	5,26%
0	7	450	38250	4	3	450	153826	9	0	298	596	1	3	450	262894	455566	0,2693	25,13%
0	7	450	38250	2	1	145	21538	9	0	298	596	1	3	450	262894	323278	0,3233	21,88%
0	7	450	38250	2	1	145	21538	0	2	145	29000	2	1	145	21538	110326	0,5023	11,33%
0	7	450	38250	2	1	145	21538	1	3	450	262894	4	4	450	88576	411258	0,4644	25,01%

Tabela 4.13 População da vigésima geração.

O melhor fitness da vigésima geração do teste acima ficou em 0,0789, com perda de material de 77692mm<sup>2</sup>, o que representa 5,26% da chapa de aço.

O teste foi repetido 100 vezes para garantir-se a aleatoriedade nos resultados, e para cada teste realizado, guardou-se o valor do melhor fitness da vigésima geração. Com os valores dos melhores fitness gerou-se um gráfico, representado na Figura 4.7, e pode-se perceber que os valores apresentam pouca variação entre eles, com isso mostra-se que mesmo que o AG não encontre a solução ótima em todos os testes, ele garante soluções muito boas.

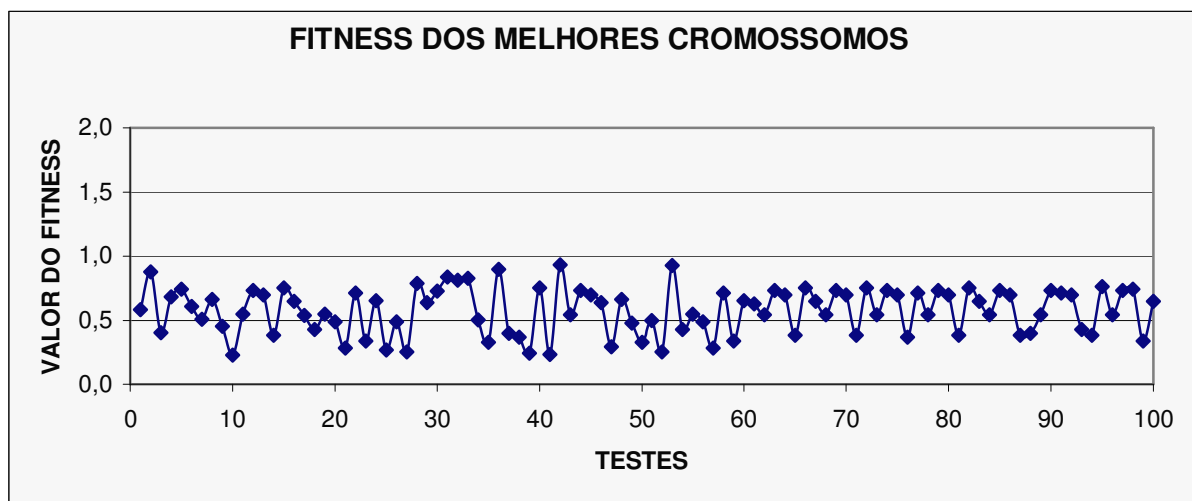


Figura 4.7 Valor dos fitness de 100 testes realizados.

Utilizando-se o mesmo banco de faixas (genes) que gerou o resultado apresentado na tabela 4.13, propõe-se a resolução do problema de programação linear inteiro com a utilização do software QM. Logo, deseja-se resolver o problema de otimização que segue em que  $x$  representa um cromossomo, que é o mesmo que um padrão de corte para

uma chapa. Assim,  $x_1$  representa o padrão 1 de corte de uma chapa,  $x_2$  representa o padrão 2 de corte de uma chapa, e segue até o trigésimo padrão de corte. A função objetivo minimiza a perda de cada um dos padrões de corte da Tabela 4.11.

$$\begin{aligned}
 \text{MIN} \quad & (596x_1 + 38250x_2 + 25132x_3 + 29000x_4 + 21538x_5 + 262894x_6 + 255432x_7 + \\
 & 262894x_8 + 299250x_9 + 197644x_{10} + 284326x_{11} + 160944x_{12} + 153482x_{13} + \\
 & 182376x_{14} + 226538x_{15} + 88576x_{16} + 153482x_{17} + 197644x_{18} + 262894x_{19} + \\
 & 262894x_{20} + 160944x_{21} + 291788x_{22} + 160944x_{23} + 153482x_{24} + 153826x_{25} + \\
 & 153482x_{26} + 132394x_{27} + 21538x_{28} + 219076x_{29} + 21538x_{30}) \\
 \text{S.A.} \quad & 9x_1 + 0x_2 + 3x_3 + 0x_4 + 2x_5 + 1x_6 + 3x_7 + 1x_8 + 0x_9 + 1x_{10} + 4x_{11} + 1x_{12} + 3x_{13} + \\
 & x_{14} + 2x_{15} + 4x_{16} + 3x_{17} + 1x_{18} + 1x_{19} + 1x_{20} + 1x_{21} + 2x_{22} + \\
 & 1x_{23} + 3x_{24} + 4x_{25} + 3x_{26} + 1x_{27} + 2x_{28} + 4x_{29} + 2x_{30} \geq 11 \\
 & 0x_1 + 7x_2 + 0x_3 + 2x_4 + 1x_5 + 3x_6 + 2x_7 + 3x_8 + 3x_9 + 4x_{10} + 1x_{11} + 2x_{12} + 1x_{13} + \\
 & 0x_{14} + 3x_{15} + 4x_{16} + 1x_{17} + 4x_{18} + 3x_{19} + 3x_{20} + 2x_{21} + \\
 & 2x_{22} + 2x_{23} + 1x_{24} + 3x_{25} + 1x_{26} + 5x_{27} + 1x_{28} + 2x_{29} + 1x_{30} \geq 14
 \end{aligned}$$

A solução exata obtida pelo QM, mostrada na Figura 4.11, é exatamente a mesma obtida com a utilização do AG, no resultado descrito na Tabela 4.13. Vale destacar que esta representa a melhor solução obtida nos 100 testes realizados com o AG, e que por conta dos sorteios realizados nas operações de cruzamento e de mutação não se pode garantir a repetição dos mesmos resultados em todos os testes, conforme visto na Figura 4.8.

(untitled) Solution																		
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	
Minimize	596,	38.250,	25.132,	29.000,	21.538,	262.894,	255.432,	262.894,	299.250,	197.644,	284.326,	160.944,	153.482,	182.376,	226.538,	88.576,	153.482,	197.644
Constraint 1	9,	0,	3,	0,	2,	1,	3,	1,	0,	1,	4,	1,	3,	4,	2,	4,	3,	
Constraint 2	0,	7,	0,	2,	1,	3,	2,	3,	3,	4,	1,	2,	1,	0,	3,	4,	1,	
Solution->	2,	2,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	

(untitled) Solution																
	x16	x17	x18	x19	x20	x21	x22	x23	x24	x25	x26	x27	x28	x29	x30	RHS
Minimize	88.576,	153.482,	197.644,	262.894,	262.894,	160.944,	291.788,	160.944,	153.482,	153.826,	153.482,	132.394,	21.538,	219.076,	21.538,	
Constraint 1	4,	3,	1,	1,	1,	1,	2,	1,	3,	4,	3,	1,	2,	4,	2,	>= 11,
Constraint 2	4,	1,	4,	3,	3,	2,	2,	2,	1,	3,	1,	5,	1,	2,	1,	>= 14,
Solution->	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	Optimal \$77.692,

Figura 4.8 Solução apresentada pelo QM.

Não se pode afirmar que o resultado obtido trata-se de um mínimo global, pois pode ser apenas de um mínimo local. De qualquer forma, a comparação entre os resultados fornecidos pelo AG e os resultados da solução exata obtido no QM, mostra uma vantagem do método heurístico, seja ela, a geração das faixas, que são os valores de entrada do método exato resolvido pelo QM.

**Segundo Teste**

O segundo teste leva em consideração o aumento do número de peças inseridas como dado inicial do AG, dessa forma, aumenta-se também o número de genes, pois caso contrário, os cromossomos tornam-se inactiváveis, já que as demandas não são atendidas. Para tal análise consideram-se os três primeiros dados da Tabela 4.14.

Dimensão da chapa mãe		Blank		Consumo diário
1100	3468	122	298	10
1100	3468	145	450	13
1100	3468	145	245	11
1100	3468	61	480	13
1100	3468	696	460	10
1100	3468	714	381	7
1100	3468	344	489	14

Tabela 4.14 Dados de entrada considerando-se as três primeiras peças.

A população inicial encontra-se na Tabela 4.15 e o resultado após a quinquagésima geração pode ser visto na Tabela 4.16.

Peças do tipo 1	Peças do tipo 2	Peças do tipo 3	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Peças do tipo 3	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Peças do tipo 3	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Peças do tipo 3	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Peças do tipo 3	Altura da faixa	Perda de material na faixa	Perda de material no cromossomo	Fitness	Percentual de perda no cromossomo					
3	0	0	122	25132	1	1	3	450	286819	1	1	4	450	251294	3	0	0	122	25132	2	0	3	298	148513	1	1	1	145	22369	79259	1,7134	43,45%
1	1	2	298	155144	1	3	1	450	227369	2	3	2	450	155488	3	1	0	298	153482	2	1	2	298	118788	1	1	3	298	119619	929890	0,5778	40,40%
3	1	0	298	153482	0	7	0	450	38250	2	2	1	450	256263	2	1	2	450	285988	2	0	5	298	77463	1	1	3	298	119619	931065	0,4086	37,71%
1	1	1	145	22369	1	2	2	450	257094	3	1	0	298	153482	1	1	4	450	251294	1	0	3	145	16569	1	1	3	298	119619	820427	0,9570	41,74%
2	0	3	298	148513	0	0	4	145	17400	1	1	3	450	286819	0	0	4	145	17400	1	1	1	145	22369	1	2	2	450	257094	749595	1,7551	41,69%
2	1	2	450	285988	3	1	0	298	153482	0	7	0	450	38250	3	0	0	122	25132	4	0	1	298	146851	1	1	4	450	251294	900997	0,6224	39,60%
9	0	0	298	596	2	3	2	450	155488	1	1	3	450	286819	2	1	2	450	285988	1	1	4	450	251294	0	3	0	450	299250	1279435	0,4396	45,64%
2	0	3	298	148513	4	0	1	298	146851	1	1	4	450	251294	0	0	4	145	17400	1	3	1	450	227369	2	2	1	450	256263	1047690	0,7754	45,53%
2	3	2	450	155488	1	0	5	298	113819	0	2	0	145	29000	1	2	2	450	257094	4	0	1	298	146851	2	2	1	450	256263	958515	0,6019	41,66%
1	3	1	450	227369	1	1	3	450	286819	0	2	0	145	29000	1	1	3	298	119619	0	3	0	450	299250	2	1	2	450	285988	1248045	1,4613	50,55%

Tabela 4.15 População inicial para a Tabela 4.14.

Peças do tipo 1	Peças do tipo 2	Peças do tipo 3	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Peças do tipo 3	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Peças do tipo 3	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Peças do tipo 3	Altura da faixa	Perda de material na faixa	Peças do tipo 1	Peças do tipo 2	Peças do tipo 3	Altura da faixa	Perda de material na faixa	Perda de material no cromossomo	Fitness	Percentual de perda no cromossomo					
9	0	0	298	596	9	0	0	298	596	0	7	0	450	38250	0	7	0	450	38250	0	0	7	245	155488	0	0	4	145	17400	250580	0,06231	12,08%
9	0	0	298	596	9	0	0	298	596	0	7	0	450	38250	0	7	0	450	38250	0	0	7	245	155488	0	0	4	145	17400	250580	0,06231	12,08%
9	0	0	298	596	9	0	0	298	596	0	7	0	450	38250	0	7	0	450	38250	0	0	7	245	155488	0	0	4	145	17400	250580	0,06231	12,08%
9	0	0	298	596	9	0	0	298	596	0	7	0	450	38250	0	7	0	450	38250	0	0	7	245	155488	0	0	4	145	17400	250580	0,06231	12,08%
9	0	0	298	596	9	0	0	298	596	0	7	0	450	38250	0	7	0	450	38250	0	0	7	245	155488	0	0	4	145	17400	250580	0,06231	12,08%
9	0	0	298	596	9	0	0	298	596	0	7	0	450	38250	0	7	0	450	38250	0	0	7	245	155488	0	0	4	145	17400	250580	0,06231	12,08%
9	0	0	298	596	9	0	0	298	596	1	1	2	298	155144	1	0	2	298	155144	1	0	3	145	16569	1	2	2	450	257094	585143	0,71663	29,76%
9	0	0	298	596	9	0	0	298	596	0	7	0	450	38250	0	7	0	450	38250	2	3	2	450	155488	1	1	3	298	119619	352799	0,10814	14,29%
9	0	0	298	596	9	0	0	298	596	1	1	2	298	155144	1	1	2	298	155144	1	0	3	145	16569	1	2	2	450	257094	585143	0,53747	29,76%

Tabela 4.16 População da quinquagésima geração.

Pode-se observar que o melhor cromossomo repete-se em sete dos dez padrões de corte da chapa, isso se deve ao fato de que acontece uma convergência para a melhor solução mantendo-se os melhores

cromossomos ao longo das gerações. Em relação às perdas geradas, verifica-se que o AG se comportou de maneira estável, reduzindo perdas e atendendo a demanda em todos os casos analisados, principalmente levando-se em consideração que a indústria faz os cálculos para a canibalização manualmente e a perda normalmente é bastante grande. Para melhor verificação da estabilidade do AG, repete-se o teste acima 100 vezes e o resultado apresenta-se na Tabela 4.17 que apresenta os fitness de cada teste realizado bem como a média desses fitness. Na seqüência, um gráfico de dispersão apresentado na Figura 4.9, plotado a partir das médias dos fitness e outro com os melhores fitness, apresentado na Figura 4.10, mostram a estabilidade dos resultados.







Figura 4.9 Gráfico da média dos fitness.

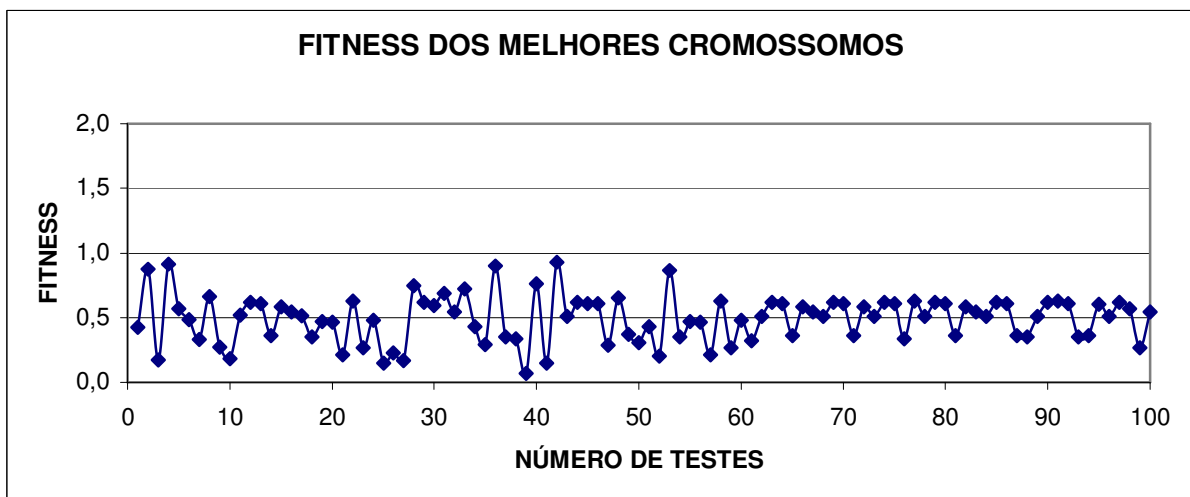


Figura 4.10 Gráfico dos melhores fitness.

Utilizando-se o mesmo banco de faixas que gerou o melhor cromossomo entre os 100 testes para o método exato tem-se o mesmo resultado, apresentado na Figura 4.11, e na Figura 4.12 o gráfico envolvendo os valores da Tabela 4.17 mostra o número de vezes que o AG converge para o ótimo, o que ocorre com 19% dos testes realizados.

Original Problem w/answers																		
(untitled) Solution																		
	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	
Minimize	596,	38.250,	20.825,	25.132,	29.000,	17.400,	197.644,	97.675,	111.326,	286.819,	154.313,	97.675,	160.944,	262.894,	287.650,	160.944,	155.488,	220.736
Constraint 1	9,	0,	0,	3,	0,	0,	1,	0,	4,	1,	2,	0,	1,	1,	0,	1,	2,	
Constraint 2	0,	7,	0,	0,	2,	0,	4,	1,	0,	1,	1,	1,	2,	3,	1,	2,	3,	
Constraint 3	0,	0,	7,	0,	0,	4,	0,	3,	2,	3,	1,	3,	0,	0,	4,	0,	2,	
Solution->	2,	2,	1,	0,	0,	1,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	

Original Problem w/answers																	
(untitled) Solution																	
	X16	X17	X18	X19	X20	X21	X22	X23	X24	X25	X26	X27	X28	X29	X30		RHS
Minimize	160.944,	155.488,	220.736,	118.788,	221.569,	185.213,	250.463,	213.276,	21.538,	285.157,	96.844,	155.144,	154.313,	97.675,	118.788,		
Constraint 1	1,	2,	2,	2,	1,	2,	2,	4,	2,	3,	1,	1,	2,	0,	2,	>=	10,
Constraint 2	2,	3,	2,	1,	2,	2,	1,	1,	1,	1,	1,	1,	1,	1,	1,	>=	13,
Constraint 3	0,	2,	2,	2,	3,	3,	3,	2,	0,	1,	2,	2,	1,	3,	2,	>=	11,
Solution->	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	Optimal	\$115.917,

Figura 4.11 Resultado do método exato no QM.

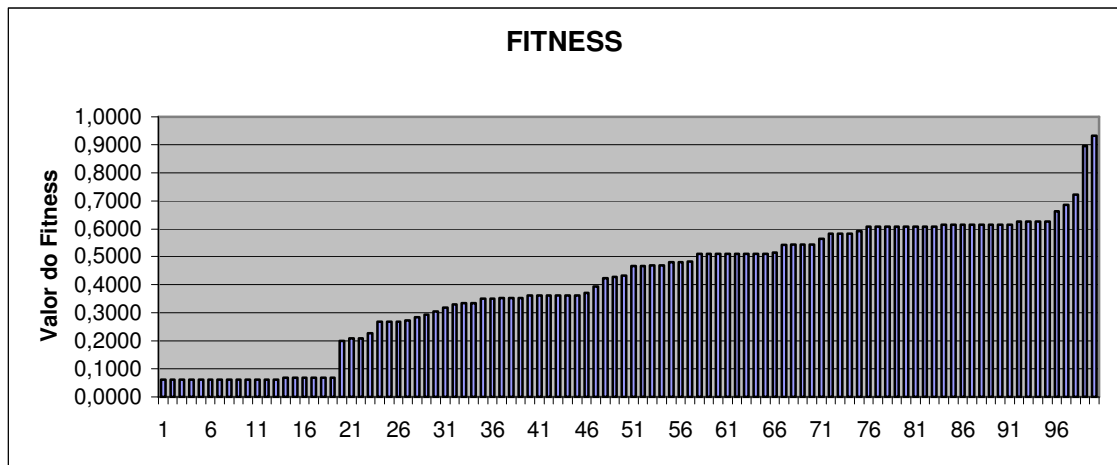


Figura 4.12 Gráfico com os melhores fitness da Tabela 4.17.

### **Terceiro Teste**

O terceiro teste verifica a influência do número de gerações nos resultados. Observa-se que o fitness reduz até a vigésima geração e depois praticamente se estabiliza, mostrando uma melhora pouco significativa, a Figura 4.13 mostra essa análise feita a partir do melhor fitness de algumas gerações do segundo teste.

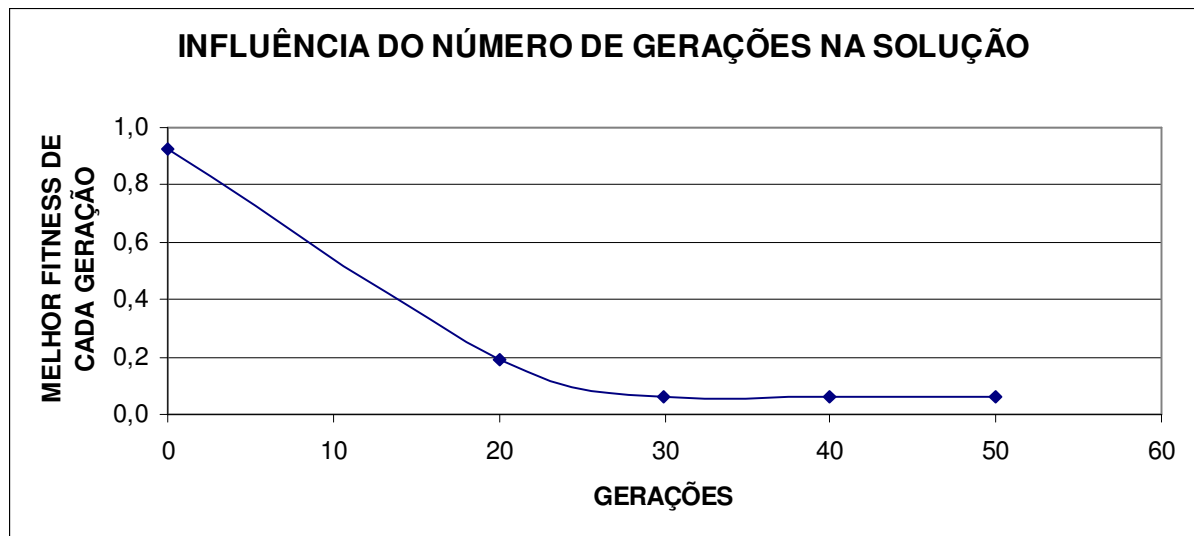


Figura 4.13 Influência do número de gerações na melhoria do fitness.

Para cada tabela de valores de entrada utilizada essa análise muda minimamente, porque em todos os casos testados, a partir da vigésima geração o cromossomo ótimo já estava definido, pois como a cada geração mantêm-se os melhores cromossomos. O racional é o de que as faixas que tem menor desperdício e conseqüentemente menor fitness, continuarão em uso e as de maior fitness serão descartadas. A prova de que isso realmente acontece é que depois de algumas gerações existiam vários cromossomos idênticos.

## 5. Conclusões.

### 5.1. Conclusões Gerais.

A seguir são apresentadas algumas considerações a respeito da aplicação do AG para corte guilhotinado bidimensional na solução de um problema de canibalização.

Constatou-se após a aplicação do AG e dos algoritmos de criação e teste das faixas que os resultados alcançados foram muito bons, gerando um desperdício pequeno e atendendo inteiramente a demanda. Os algoritmos são fáceis de serem utilizados, adaptáveis para outras quantidades e tamanho de peças e chapas e para qualquer quantidade de padrões e número de gerações. Dessa forma atingiu-se o objetivo inicial do trabalho conseguindo-se bons padrões de corte de chapas, e o principal: de forma extremamente rápida.

O método heurístico aplicado apresentou ganhos reais no desperdício, mesmo que não se tenha garantia que o AG não gera mínimo global, pois podemos ter chegado apenas em um mínimo local, e para a canibalização onde o desperdício normalmente é bastante grande, a vantagem obtida foi muito boa.

Evidentemente que, para demandas maiores, tanto a geração de faixas, quanto o tamanho dos cromossomos devem ser readaptados, talvez gerando setups para cada faixa ou gene de forma que estes possam ser definidos dentro dos próprios cromossomos, sem alterar o número de faixas

contidas em cada padrão de corte da chapa. Não foram testados nesse trabalho os fitness (perdas e atendimento da demanda) para uma repetição em escala maior do mesmo cromossomo, e que certamente não seria a melhor saída para um aumento explosivo na demanda, o que não é o caso da canibalização.

Como sugestão para trabalhos futuros, fica a possibilidade de adaptação deste método para um número maior de peças, não visando apenas atender o problema de canibalização, mas a demanda de todos os cortes de chapa realizados na fábrica.

## REFERÊNCIAS

ALVARENGA, Arlindo Gomes; PARADA Victor. **Um Método de Busca Informada para o Problema do Corte de Peças**. Universidad de Concepción- Concepción-Chile. 1990.

ALVARENGA, Arlindo Gomes; PARADA Victor, CARNEIRO, Sebastião A. **Problema de Corte via Algoritmo Genético**. Universidad de Concepción- Concepción-Chile. 1990.

ALVARENGA, Arlindo Gomes; DAZA, Victor Parada **Exact solutions for constrained two-dimensional cutting problems**, European Journal of Operational Research 84, 633-644,1995.

ARAUJO, Olinto César Bassi. **Problema de Corte e Empacotamento Tridimensional e Integração com Roteamento de Veículos**. Tese de Doutorado. Campinas, SP, 2006.

ARENALES, M. N; YANASSE, H.H; LORENA, L.A.N; FRANÇA, P.M; VIANNA, A.C.G; MORABITO, R; PUREZA, V.M; RANDAL, F. INPE **Projeto Cortes e Empacotamento Assistido Por Computador**. São Paulo, 1997.

BARCELLOS, J.C.H., **Algoritmos Genéticos Adaptativos**: um estudo comparativo. Dissertação de Mestrado. São Paulo, EPUSP, 2000.

CHERRI, A. C.;ARENALES, M. N.. **O Problema de corte de estoque com reaproveitamento das sobras de material** - Heurística FFD Modificada. In: Simpósio Brasileiro de Pesquisa Operacional, 2005, Gramado RS. Anais do Simpósio Brasileiro de Pesquisa Operacional,. Rio de Janeiro : SBPO, 2005. v. 1. p. 1-12.

CHRISTOFIDES, Nicos. HADJICONSTATINOU. **An Exact Algorithm for Orthogonal 2-D Cutting Problems Using Guillotine Cuts**. European Journal of Operational Research 83, 21-38, 1995.

DAZA, V.P; ALVARENGA, A. G.;DIEGO J. **Exact Solutions For Constraint Two - dimensional cutting Problems European Journal of Operational Research 83, 633-644. 1995.**

GAZANGI, R. R.; FIGUEIREDO, M. F.. **“Simultaneous Emergence of Conflicting Basic Behaviors and Their Coordination in an Evolutionary Autonomous Navigation System”** In: **2002 World Congress on Computational Intelligence - Congress on Evolutionary Computing**, Honolulu, Hawaii, EUA, 2002.

GILMORE, P.C., GOMORY, R.E.(1963), **Multistage cutting stock problems of two and more dimensions**, Operations Research, Vol. 13, pp. 94-120.

GOLDENBERG, D. E., **Genetic Algorithms in Search, Optimization and Machine Learning**. USA: Addison Wesley Publishing Company, 1989.

GRAMANI, Maria Cristina Nogueira. **Otimização do Processo de Cortagem Acoplado ao Planejamento da Produção**. Tese de Doutorado. Campinas, SP, 2001.

HERZ, J.C. **Recursive Computational Procedure for Two-Dimensional Stock Cutting**, IBM Journal of Research and Development 16, pp. 462-469. 1972.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. Ann Arbor: The University Of Michigan Press, 1975.

LOPES, M. A; ARENALES S.H. **Otimização: O problema de Corte de Peças Bidimensional**. Congresso Nacional de Matemática Aplicada e Computacional 2003.

MIRANDA, M. N. ; LIMA R.N.B. ; PEDROZA, A. C. P. ; MESQUITA FILHO, A. C. . **HW/SW Codesign of Protocols Based on Performance Optimization Using Genetic Algorithms**. In: J. Moreira de Souza; Nelson L.S. Fonseca; E.A. De Souza e Silva. (Org.). *Teletraffic Engineering in the Internet Era*. 1ª ed. Elsevier Science B.V., 2001, v. 4, p. 259-269.

MITCHELL, M. **An Introduction to Genetic Algorithms**. The MIT Press, EUA, 1998.

MORABITO, R; ARENALES, M. **An and/Or-Graph approach to the solution of two dimensional non-guillotine cutting problems**, European Journal of Operational Research 84, 599-617. 1995

MORABITO, R; ARENALES, M. **Stage and Constrained two - dimensional guillotine cutting problems: an AND/OR-graph approach**, European Journal Of Operational Research 94, 548-560.1996.

OLIVEIRA, J; FERREIRA, J. **An improved version of wang's Algorithm for two dimensional cutting problems**, European Journal of Operational Research 44, 256-266.1990.

OLIVEIRA, S. R., **O Algoritmo Genético na Otimização do Projeto Estrutural de Embarcações**. Tese de Doutorado. São Paulo: EPUSP, 1998.

PILEGGI, G. C. F. ; MORABITO, R. ; ARENALES, M. N. . **Heurísticas para os Problemas de Geração e Sequenciamento de Padrões de Corte Bidimensionais**. Pesquisa Operacional, v. 27, p. 549-568, 2007.

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



SILVA, Rociro T. **Algoritmos Genéticos e Programação Linear Inteira Aplicados ao Problema de Corte Unidimensional**. Monografia de conclusão de curso. Curitiba, PR, 2005.

VASKO, Francis J. (1989), **An Computational Improvement to Wang's two dimensional Cutting Stock Algorithm**, Computers ind. Engng 16 109-115. 1989.

VIANNA, Andréa.C.G, ARENALES, M.N.e GRAMANI, Maria C.N. **Two-stage and constraint two-dimensional guillotine cutting** .VI Oficina de Problemas de Cortes e Empacotamento Campinas,SP. 2002.

VIANA, Valdisio. **Meta-Heurísticas e Programação Paralela em Otimização Combinatória**. (UFC) Universidade Federal do Ceará. Fortaleza 1998.

TEODORO, Alan agosto. **O Problema de Corte Bidimensional: Uma abordagem utilizando o método de geração de colunas**. Dissertação de Mestrado. Campinas, SP, 2003.

WANG, P.Y.(1983).**Two Algorithm for constrained two-dimensional cutting Stock Problems**. Operational Research 31, 573-587.1983

## ANEXO 1 – Algoritmo FAIXAS

```

n=1 (contador)
LC=1130 (largura da chapa)
NPC=14 (definição do número de planos de corte)
NP=2 (número de peças a serem cortadas)
Saída=zeros(NPC,NP+2) (dimensão da matriz saída de dados)
DP=[697 322 1; 298 122 2; 322 697 1; 122 298 2] (dimensão das peças na horizontal e vertical)
posx=0 (posição da peça na chapa, no sentido horizontal)
posy=0 (posição da peça na chapa, no sentido vertical)
para k=1:2*NP (considerando a rotação da peça)
    areaplacas=0 (a faixa está vazia, portanto area zero)
    figure
    posx=0;
    rectangle('Position',[0,0,LC,DP(k,2)]) (encaixa a peça no retângulo )
    enquanto (LC-posx)>=DP(k,1) (verifica se encaixa mais uma peça na largura da chapa)
        rectangle('Position',[posx,0,DP(k,1),DP(k,2)],'FaceColor','r ') (gráfico)
        se k<=NP
            saida(k,k)=saida(k,k)+1
        senão
            saida(k,k-NP)=saida(k,k-NP)+1 (organização da matriz saída)
        fim
        posx=posx+DP(k,1)
    fim
    para j=1:NP
        areaplacas=areaplacas+saida(k,j)*DP(k,1)*DP(k,2) (cálculo da área usada pelas peças)
    fim
    saida(k,NP+1)=DP(k,2) (a altura da faixa será a altura da maior peça usada na construção da
faixa e ocupará a posição j-1 da matriz de saída)
    saida(k,NP+2)=LC*DP(k,2)-areaplacas () (a área desperdiçada, em milímetros quadrados,
ocupará a j-ésima posição na matriz de saída)
    fim
para k=1:(NPC-2*NP)
    areaplacas=0
    n=1
    altura=0
    posx=0
    figure
    rectangle('Position',[0,0,LC,1])
    enquanto n==1

```

```

peca=randperm(2*NP )(criação de faixas por sorteio)
peca=peca(1)
enquanto (LC-posx)>=DP(peca,1)
    rectangle('Position',[posx,0,DP(peca,1),DP(peca,2)],'facecolor','r')
    saida(k+2*NP,DP(peca,3))=saida(k+2*NP,DP(peca,3))+1
    altura=[altura DP(peca,2)]
    alturafaixa=max(altura)
    posx=posx+DP(peca,1)
    contador=0
    peca=randperm(2*NP)
    peca=peca(1) (verifica se alguma peça ainda encaixa na faixa)
    para i=1:2*NP
        se (LC-posx)>DP(i,1)
            contador=contador+1
        senão
            contador=contador
        fim
    fim
    se contador==0
        n=0
    fim
fim
fim
parar j=1:NP
    areaplacas=areaplacas+saida(k+2*NP,j)*DP(j,1)*DP(j,2)
fim
saida(k+2*NP,NP+1)=alturafaixa;
saida(k+2*NP,NP+2)=LC*alturafaixa-areaplacas;
fim

```

## ANEXO 2 – Algoritmo TESTE

```

clc
populacao = [];
populacao_descartada= []; (na populacao_descartada estão os individuos inactivos)
saida
D=[10; 13; 11] (demanda)
n=10; (número de soluções factíveis)
ng=4; (número de gerações)
NPM=size(saida,1); (%número de padrões)
NC=size(saida,2);
while size(populacao,1)~=n
sol=[];
  for i=1:ng
    v=randperm(NPM);
    l=v(1);
    sol= [sol saida(l,:)];
  end
  sol(:,ng*NC+1)=0;
  for i=1:ng
    sol(:,ng*NC+1)=sol(:,ng*NC+1)+sol(:,size(saida,2)*i);
  end
  peca=[];
  for j=1:(size(saida,2)-2)
    peca(j,1)=sol(:,j);
    for i=1:ng-1
      peca(j,1)=peca(j,1)+sol(:,j+i*size(saida,2));
    end
  end
  for i=1:size(peca,1)
    peca(i,1)=peca(i,1)/D(i,1);
  end
  PAD=prod(peca);
  SdA=0;
  for i=1:ng
    SdA=SdA+sol(:,size(saida,2)*i-1);
  end

  sol(:,ng*NC+2)=((sol(:,ng*NC+1))/(LPM*SdA))/PAD);

```

```
populacao = [populacao; sol];
```

```
plot(populacao(:,ng*NC+1),populacao(:,ng*NC+2),'m+')
```

```
end % do while
```

```
populacao(:,1:end-1)
```

```
populacao(:,end)
```

## ANEXO 3 – Algoritmo Genético – AG

```
faixas
saida
teste
ngeracoes=0
while ngeracoes<50
    ngeracoes=ngeracoes+1
    sorteio_pais_h
    cruzamento_h
    mutacao_h
    sorteio_final

if ngeracoes==10
plot(populacao(:,ng*NC+1),populacao(:,ng*NC+1),'k+')
end
end
```

## ANEXO 4 – Algoritmo SORTEIO\_PAIS

```
numero=floor(size(populacao,1)*.6); (60% do total da populacao (floor arredonda para baixo))
sorteio=randperm(size(populacao,1));
for cont=1:numero
    pais(cont,:)=populacao(sorteio(cont),:);
end
```

## ANEXO 5 – Algoritmo CRUZAMENTO

```

nc=size(pais,2);
filhos=zeros(size(pais));
for cont=1:2:(size(pais,1)-1)
    sorteio2=randperm(ng-1);
    filhos(cont,1:(sorteio2(1)*NC))=pais(cont,1:(sorteio2(1)*NC));
    filhos(cont,((sorteio2(1)*NC+1):(nc-2)))=pais(cont+1,((sorteio2(1)*NC+1):(nc-2)));
    filhos(cont+1,1:(sorteio2(1)*NC))=pais(cont+1,1:(sorteio2(1)*NC));
    filhos(cont+1,((sorteio2(1)*NC+1):(nc-2)))=pais(cont,((sorteio2(1)*NC+1):(nc-2)));
end
%Soma das areas
filhos(:,ng*NC+1)=0;
for i=1:ng
    filhos(:,ng*NC+1)=filhos(:,ng*NC+1)+filhos(:,size(saida,2)*i);
end
%Calculo do fitness
for cont=1:size(filhos,1)
    peca=[];
    for i=1:(size(saida,2)-2)
        peca(i,1)=filhos(cont,i);
        for j=1:ng-1
            peca(i,1)=peca(i,1)+filhos(cont,i+j*size(saida,2));
        end
    end
    for i=1:size(peca,1)
        peca(i,1)=peca(i,1)/D(i,1);
    end
    PAD=prod(peca);
    SdA=0;

    for i=1:ng
        SdA=SdA+filhos(cont,size(saida,2)*i-1);
    end
    filhos(cont,ng*NC+2)=((filhos(cont,ng*NC+1))/(LPM*SdA))/PAD;
end

```



## ANEXO 6 – Algoritmo MUTAÇÃO

```

nova_populacao=[populacao; filhos];
numero_mutacoes=ceil(size([populacao; filhos],1)*.01); (Porcentagem de individuos mutados)
mutado1=randperm(size(nova_populacao,1));
gene_alterado=[];
for cont=1:numero_mutacoes
mutado(cont)=mutado1(cont);
partemutada=randperm(2);
gene_alterado1=randperm(size(saida,1));
for cont1=1:ng/2
gene_alterado=[gene_alterado saida(gene_alterado1(cont1,:))];
end
if partemutada==1
nova_populacao(mutado(cont),1:(nc-2)/2)=gene_alterado;
else
nova_populacao(mutado(cont),((nc-2)/2+1):(nc-2))=gene_alterado;
end
%SOMA DAS AREAS
nova_populacao(mutado(cont),ng*NC+1)=0;
%Calculo Fitness
for i=1:ng

nova_populacao(mutado(cont),ng*NC+1)=nova_populacao(mutado(cont),ng*NC+1)+nova_populacao
(mutado(cont),size(saida,2)*i);
end
for i=1:(size(saida,2)-2)
peca(i,1)=nova_populacao(mutado(cont),i);
for j=1:ng-1
peca(i,1)=peca(i,1)+nova_populacao(mutado(cont),i+j*size(saida,2));
end
end

for i=1:size(peca,1)
peca(i,1)=peca(i,1)/D(i,1);
end
PAD=prod(peca);
SdA=0;

```

```
for i=1:ng
    SdA=SdA+nova_populacao(mutado(cont),size(saida,2)*i-1);
end

nova_populacao(mutado(cont),ng*NC+2)=((nova_populacao(mutado(cont),ng*NC+1)/(LPM*SdA))/PA
D);;
end
```

## ANEXO 7 – Algoritmo NOVA\_POPULAÇÃO

```

nova_populacao2=nova_populacao;
for cont=1:ceil(n*.5)
    [minimo indice(cont)]=min(nova_populacao2(:,end));
    populacao_final(cont,:)=nova_populacao2(indice(cont),:);
    nova_populacao2(indice(cont),end)=inf;
end
i=1;
cont=1;
while i<=size(nova_populacao2,1)
    disparo=0;
    for j=1:size(indice,2)
        if i==indice(j)
            disparo=1;
        end
    end
    if disparo==0
        nova_populacao3(cont,:)=nova_populacao2(i,:);
        cont=cont+1;
        i=i+1;
    else
        i=i+1;
    end
end
i=1;
sorteio=randperm(size(nova_populacao3,1));
for cont=(ceil(n*.5)+1):n
    populacao_final(cont,:)=nova_populacao3(sorteio(i),:);
    i=i+1;
end
populacao_final(:,1:end-2)
populacao_final(:,end-1)
populacao_final(:,end)
populacao=populacao_final;

```