

**UNIVERSIDADE NOVE DE JULHO - UNINOVE**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

**META-HEURÍSTICAS BIOINSPIRADAS APLICADAS AO PROBLEMA DO**  
**CORTE BIDIMENSIONAL GUILHOTINADO EM UMA INDÚSTRIA VIDREIRA**

**FLAVIO MOREIRA DA COSTA**

**SÃO PAULO**

**2014**

**FLAVIO MOREIRA DA COSTA**

**META-HEURÍSTICAS BIOINSPIRADAS APLICADAS AO PROBLEMA DO  
CORTE BIDIMENSIONAL GUILHOTINADO EM UMA INDÚSTRIA VIDREIRA**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Nove de Julho como parte dos requisitos exigidos para a obtenção do grau de Mestre em Engenharia de Produção.

Prof. Renato José Sassi, Dr.- Orientador, UNINOVE

**SÃO PAULO**

**2014**

Costa, Flavio Moreira da.

Meta-heurísticas bioinspiradas aplicadas ao problema do corte bidimensional guilhotinado em uma indústria vidreira. / Flavio Moreira da Costa. 2014.

90 f.

Dissertação (mestrado) – Universidade Nove de Julho - UNINOVE, São Paulo, 2013.

Orientador (a): Prof. Renato José Sassi.

1. Algoritmo Genético. 2. Algoritmo Colônia de Formigas. 3. Corte bidimensional guilhotinado. 4. Meta-heurísticas bioinspiradas.

I. Sassi, Renato José. II. Título

CDU 658.5

**META-HEURÍSTICAS BIOINSPIRADAS APLICADAS AO PROBLEMA DO  
CORTE BIDIMENSIONAL GUILHOTINADO EM UMA INDÚSTRIA VIDREIRA**

Por

**FLAVIO MOREIRA DA COSTA**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Nove de Julho como parte dos requisitos exigidos para a obtenção do grau de Mestre em Engenharia de Produção:

---

Presidente: Prof. Dr. Renato José Sassi - Orientador, UNINOVE

---

Membro: Prof. Dr. Nilton Cesar Furtado Canto, UNINOVE

---

Membro: Prof. Dr. Leonardo Junqueira, UNINOVE

**SÃO PAULO**

**2014**

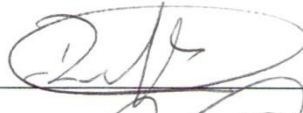
São Paulo, 23 de janeiro de 2014.

**TERMO DE APROVAÇÃO**

Aluno: Flavio Moreira da Costa

Título da Dissertação: META-HEURÍSTICAS BIOINSPIRADAS APLICADAS AO PROBLEMA DO CORTE BIDIMENSIONAL GUILHOTINADO EM UMA INDÚSTRIA VIDREIRA.

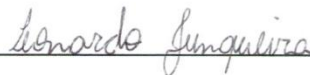
Presidente: PROF. DR. RENATO JOSÉ SASSI



Membro: PROF. DR. NILTON CESAR FURTADO CANTO



Membro: PROF. DR. LEONARDO JUNQUEIRA



Dedico este trabalho ao meu pai Dioges Batista da Costa e à minha mãe Luzia Moreira da Costa, por todo esforço, incentivo e pela sólida base moral, sem a qual eu não teria alçado voos tão altos.

## AGRADECIMENTOS

Agradeço primeiramente a Deus, autor e mantenedor da vida, por ser o meu refúgio nos momentos de adversidade e por me conceder sabedoria em todos os momentos da minha jornada.

Ao Professor Dr. Renato José Sassi pelo incentivo a pesquisa, pelo empenho, paciência e orientação.

Agradeço à minha esposa Renata Silva Pereira da Costa, pela compreensão e paciência durante o período do mestrado.

Agradeço aos docentes do curso de Mestrado em Engenharia de Produção e funcionários da Universidade Nove de Julho. Em especial ao Professor Dr. Fabio Henrique Pereira, por compartilhar sua experiência, pelas contribuições e incentivo.

Aos membros da banca: Prof. Dr. Nilton Cesar Furtado Canto e Prof. Dr. Leonardo Junqueira.

Agradeço os colegas do nosso grupo de pesquisa e ao meu amigo Flávio Grassi, pela parceria durante o Mestrado.

À Universidade Nove de Julho pela bolsa de estudos no Programa de Pós-Graduação em Engenharia de Produção.

À CAPES pela bolsa PROSUP.

“Bem-aventurado o homem que acha sabedoria, e o homem que adquire conhecimento.” Provérbios 3:13.



## RESUMO

O corte bidimensional guilhotinado é um problema de otimização combinatória que consiste em determinar um arranjo de itens a serem cortados a partir de um objeto maior, maximizando a utilização do material, porém respeitando as restrições do equipamento de corte e do fluxo de produção. A otimização do corte bidimensional guilhotinado é um importante fator no desempenho dos sistemas de produção das indústrias vidreiras, já que possibilita uma melhor utilização dos materiais utilizados. Pesquisas demonstram que as meta-heurísticas bioinspiradas como Algoritmos Genéticos e Algoritmo Colônia de Formigas podem ser aplicadas na solução de problemas que envolvem otimização combinatória, como o problema do corte bidimensional guilhotinado. O Algoritmo Genético é uma abstração de modelos de evolução presentes na natureza que opera sobre uma população de indivíduos, por meio da aplicação de mecanismos de seleção, cruzamento e mutação, gerando novos indivíduos que, a cada geração, tornam-se mais aptos. O Algoritmo Colônia de Formigas é inspirado no comportamento de colônias de formigas que são capazes de encontrar o caminho mais curto entre suas colônias e as fontes de alimento. Influenciadas pela presença de feromônios no caminho, as formigas tendem a seguir na direção em que a concentração de feromônios é mais forte. Este trabalho teve como objetivo aplicar meta-heurísticas bioinspiradas, especificamente o Algoritmo Genético e o Algoritmo Colônia de Formigas, individualmente e combinados, ao problema do corte bidimensional guilhotinado em uma indústria vidreira. Foi utilizada na realização dos experimentos uma base de dados real com pedidos de corte fornecidos por uma indústria vidreira da cidade de São Paulo e 500 instâncias obtidas da literatura divididas em 10 classes com tamanhos variados de itens e objetos. Os melhores resultados foram comparados com dois *softwares* comerciais de otimização do corte bidimensional. Os resultados finais foram satisfatórios, o que confirma as meta-heurísticas bioinspiradas como uma opção para solucionar o problema do corte bidimensional guilhotinado em uma indústria vidreira.

**Palavras-chave:** Algoritmo Genético. Algoritmo Colônia de Formigas. Corte Bidimensional Guilhotinado. Meta-heurísticas Bioinspiradas.

## ABSTRACT

The two-dimensional guillotined cutting is a combinatorial optimization problem that consists in determining an arrangement of items to be cut from a larger piece, maximizing the material use, but respecting the restrictions imposed by the cutting equipment and the production flow. The optimization of two-dimensional guillotined cutting is an important factor for production systems performance at glassworks industries, because it maximizes the materials use. Several research have shown the application of bio-inspired metaheuristics like Genetic Algorithm and Ant Colony Algorithm in solving combinatorial optimization problems as two-dimensional guillotine cutting. The Genetic Algorithm is an abstraction of natural evolution models which operate on a population of individuals, through the application of mechanisms of selection, crossover and mutation, generating new and fitter individuals in each generation. The Ant Colony Algorithm is inspired in ant colonies behavior, that are capable of finding the shortest path between their colony and food sources. Influenced by pheromones presence in this way, these ants tend to follow the direction in which the concentration of pheromone is stronger. This work aimed to apply, individually and combined, the Genetic Algorithm and Ant Colony Algorithm to solve the two-dimensional guillotine cutting problem in a glass industry. For the experiments it was used a real-world database of 50 requests provided by a cut glass industry in São Paulo city and 500 instances, obtained from literature, divided into 10 classes with varying sizes of items and objects. The best results were compared with two commercial softwares focused on two-dimensional cutting optimization. The final results were satisfactory, confirming the bio-inspired metaheuristics as an option to solve the two-dimensional guillotine cutting problem in a glass industry.

**Keywords:** Genetic Algorithms. Ant Colony Optimization. Two-dimensional Guillotined Cutting. Bio-inspired Metaheuristics.

## SUMARIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
1.1	JUSTIFICATIVA E MOTIVAÇÃO	18
1.2	PROBLEMA DA PESQUISA	18
1.3	OBJETIVO GERAL	18
1.3.1	Objetivos Específicos	19
1.4	ORGANIZAÇÃO DO TRABALHO	19
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>20</b>
2.1	A INDÚSTRIA VIDREIRA	20
2.2	PROBLEMAS DE ALOCAÇÃO DE RECURSOS	24
2.3	O PROBLEMA DO CORTE BIDIMENSIONAL GUILHOTINADO NA INDÚSTRIA VIDREIRA	25
2.4	OTIMIZAÇÃO COMBINATÓRIA E PROBLEMAS DE SATISFAÇÃO DE RESTRIÇÕES	30
2.5	HEURÍSTICAS E META-HEURÍSTICAS	33
2.6	COMPUTAÇÃO BIOINSPIRADA	35
2.6.1	Algoritmos Genéticos	36
2.6.2	Algoritmo de Otimização por Colônia de Formigas	43
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>47</b>
3.1	CARACTERIZAÇÃO METODOLÓGICA	47
3.2	FERRAMENTAS, PLATAFORMA DE ENSAIO E BASE DE DADOS	48
3.3	METODOLOGIA EXPERIMENTAL	48
<b>4</b>	<b>APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS</b>	<b>62</b>
4.1	REALIZAÇÃO DOS EXPERIMENTOS	62
<b>5</b>	<b>CONCLUSÕES E PERSPECTIVAS FUTURAS</b>	<b>76</b>
5.1	CO-ORIENTAÇÃO EM PROJETO DE INICIAÇÃO CIENTÍFICA	78
5.2	PUBLICAÇÕES DO AUTOR	79
	REFERÊNCIAS	80
	ANEXOS	86
	APÊNDICES	89

## LISTA DE FIGURAS

Figura 1 – Vista panorâmica da Providro e da terraplenagem para a construção da Cebrace II, que seria inaugurada em 1989. ....	22
Figura 2 – Fases do processo de fabricação do vidro <i>float</i> . ....	23
Figura 3 – Itens encaixados a partir da heurística <i>Next-fit</i> . ....	35
Figura 4 – Fluxograma com a sequência padrão de execução de um AG. ....	38
Figura 5 – Exemplo do processo de cruzamento entre cromossomos com codificação baseada em ordem. ....	41
Figura 6 – Formação de cromossomos inválidos devido a elementos repetidos ....	41
Figura 7 – Exemplo da aplicação do operador de <i>crossover</i> de dois pontos para cromossomos com codificação baseada em ordem ....	42
Figura 8 – Tendência das formigas em seguir o trajeto mais curto em função da concentração de feromônios ....	44
Figura 9 – Exemplo de sequencia (solução, indivíduo ou formiga) gerada pela meta-heurística ....	50
Figura 10 - Iteração entre as meta-heurísticas e heurísticas construtivas ....	50
Figura 11 – Processo de alocação ou encaixe dos itens nos objetos, através da aplicação das heurísticas construtivas ....	51
Figura 12 – Exemplo de aplicação da roleta viciada. ....	53
Figura 13 – Comparação de duas solução a partir da função objetivo da Equação 7 ....	54
Figura 14 – Fluxograma das fases do processo de combinação cooperativa entre as meta-heurísticas ....	57
Figura 15 – Aplicação da heurística construtiva FC ....	59
Figura 16 – Aplicação da heurística KP ....	60
Figura 17 – Validação da implementação das heurísticas BFDWDH e FFDWDH. ....	63
Figura 18 – Calibração do AG em função do número de gerações. ....	64

## LISTA DE QUADROS

Quadro 1 – Breve cronologia da indústria vidreira no Brasil .....	21
Quadro 2 – Processos industriais e de produção que envolvem alocação de recursos .....	24
Quadro 3 – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 1 .....	69
Quadro 4 – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 2 .....	70
Quadro 5 – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 3 .....	70
Quadro 6 – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 4 .....	70
Quadro 7 – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 5 .....	71
Quadro 8 – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 6 .....	71
Quadro 9 – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 7 .....	71
Quadro 10 – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 8 .....	72
Quadro 11 – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 9 .....	72
Quadro 12 – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 10 .....	72
Quadro 13 – Comparação das médias de cada classe e da média geral do AG+ACO com as heurísticas FCRG e KPRG .....	73
Quadro 14 – Comparação dos resultados (quantidade de chapas utilizadas) da aplicação do Lisec GPS.opt e do AG+ACO nos problemas P5 a P12 .....	74
Quadro 15 – Comparação dos resultados (quantidade de chapas utilizadas) da aplicação do Opty Way e do AG+ACO nos problemas P13 a P20 .....	74

## LISTA DE TABELAS

Tabela 1 – Informações dos problemas P1, P2, P3 e P4 .....	52
Tabela 2 – Informações dos problemas P5 a P12 .....	60
Tabela 3 – Informações dos problemas P13 a P20 .....	61
Tabela 4 - Resultados da aplicação do AG nos problemas P1, P2, P3 e P4 .....	64
Tabela 5 - Resultados da aplicação do ACO nos problemas P1, P2, P3 e P4 .....	65
Tabela 6 - Resultados da aplicação das meta-heurísticas combinadas nos problemas P1, P2, P3 e P4 .....	66
Tabela 7 - Comparação dos resultados de cada fase em relação à quantidade de objetos necessários para cada problema .....	66
Tabela 8 - Comparação dos resultados de cada fase em relação ao tempo de processamento de cada problema .....	67
Tabela 9 - Comparação dos resultados das heurísticas individuais em relação às mesmas como auxiliares das meta-heurísticas combinadas .....	68
Tabela 10 - Comparação dos resultados das meta-heurísticas individuais em relação as mesmas combinadas, após os ajustes da Fase 5 .....	68
Tabela 11 - Comparação do tempo de processamento das meta-heurísticas individuais em relação as mesmas combinadas .....	68

## LISTA DE ABREVIATURAS E SIGLAS

ACO	-	<i>Ant Colony Optimization</i>
AG	-	Algoritmo Genético
AS	-	<i>Ant System</i>
BF	-	<i>Best-fit</i>
BFDWDH	-	<i>Best-fit Decreasing Width Decreasing Height</i>
BPP	-	<i>Bin Packing Problem</i>
CNC	-	Controle Numérico Computadorizado
FF	-	<i>First-fit</i>
FFDWDH	-	<i>First-fit Decreasing Width Decreasing Height</i>
IA	-	Inteligência Artificial
PO	-	Pesquisa Operacional
MMAS	-	<i>MAX-MIN Ant System</i>
NF	-	<i>Next-fit</i>
PC	-	Problema de Corte
PCB	-	Problema de Corte Bidimensional
PCBG	-	Problema de Corte Bidimensional Guilhotinado
PCBNG	-	Problema de Corte Bidimensional Não-Guilhotinado
PCE	-	Problema de Corte e Empacotamento
PE	-	Problema de Empacotamento
PG	-	Programação Genética
PSO	-	<i>Particle Swarm Optimization</i>
PSR	-	Problemas de Satisfação de Restrições
SC	-	Sistemas Classificadores
WF	-	<i>Worst-fit</i>

## 1 INTRODUÇÃO

A fabricação do vidro é uma técnica muito antiga. São inúmeros os povos que reclamam o privilégio da descoberta e fabricação do vidro em épocas remotas da antiguidade, dentre eles os fenícios, egípcios, persas, romanos, chineses, dentre outros.

Plínio, o grande naturalista romano, nascido no ano 23 de nossa era, em sua enciclopédia *Naturalis Historia* atribui aos fenícios a obtenção dos vidros. Segundo o relato, ao desembarcarem nas costas da Síria há cerca de 7000 anos A.C., os fenícios improvisaram fogões usando blocos de salitre sobre a areia e observaram que, passado algum tempo de fogo vivo, escorria uma substância líquida e brilhante que se solidificava rapidamente. Os fenícios dedicaram muito tempo à reprodução deste fenômeno, chegando à obtenção de materiais utilizáveis (ALVES et al., 2001).

Somente no início da era Cristã as técnicas de fabricação de vidro começaram a se desenvolver, quando os romanos começaram a difundir a utilização de moldes e o sopro na fabricação do vidro. Tais técnicas de fabricação, herdadas dos sírios, possibilitaram o início da produção seriada, cujo apogeu se deu por volta do século XIII e até o século XIX, a produção do vidro ainda era considerada uma arte destinada a poucos, apesar de a indústria vidreira ter acompanhado a evolução industrial da fabricação em série ocorrida no século XX (ASSIS; TEIXEIRA, 2001).

Devido ao surgimento de novos métodos de produção, como o sistema *float*, a indústria vidreira encontra-se em nível maior de qualidade, eficiência e produtividade.

O vidro *float* é um vidro de silicato sodocálcico, plano, transparente, incolor ou colorido em sua massa, de faces paralelas e planas, que se obtém por fundição contínua e solidificação no interior de um banho de metal fundido. É comercializado para uso na arquitetura e decoração, onde os requisitos de qualidade visual são distintos para cada aplicação (NBR NM 294, 2004).

No Brasil, a indústria vidreira é relativamente nova. A primeira fábrica de vidros surgiu em Salvador, na Bahia, e foi aberta em 1810, pelo vidreiro português Francisco Ignácio de Siqueira, tendo sido pioneira na produção de vidros planos eocos no Brasil.

Segundo Assis e Teixeira (2001), a indústria vidreira no Brasil teve um crescimento lento, tomando contornos bem definidos somente a partir de 1960, devido ao fornecimento para a indústria automotiva. A partir daí, tem evoluído aos poucos para uma indústria moderna e eficiente que, no entanto, ainda tem muito a evoluir.



A pesquisa e o desenvolvimento de materiais e processos na indústria vidreira no Brasil ainda são, em muitos casos, baseados em valores empíricos e a evolução dos processos industriais depende do acúmulo de conhecimento dos engenheiros envolvidos no processo vidreiro. Toda a tecnologia de ponta utilizada atualmente pela indústria vidreira no Brasil, bem como os principais softwares de otimização de corte, vem importada de países europeus.

Os processos produtivos da indústria vidreira, assim como de outros ramos da indústria, apesar de dispor de um maquinário moderno, ainda possuem diversos problemas relacionados à alocação de recursos.

Os problemas de alocação de recursos, geralmente impactam no desempenho dos sistemas de produção industrial. Eles podem ser encontrados nos mais diversos processos industriais e de produção, tais como: o planejamento e controle de produção (CARVALHO et al., 1998), alocação de mão de obra (PEDRO et al., 2012), o corte de materiais (BELOV; SCHEITHAUER, 2006) e escalonamento de tarefas (ROCHA, 2011), etc.

O problema de corte consiste na determinação de padrões de corte de unidades de materiais (objetos) de maneira a produzir um conjunto de unidades menores (itens), satisfazendo determinadas restrições.

O potencial das aplicações práticas da resolução de problemas de corte e problemas de alocação de recursos em geral, bem como as dificuldades para obtenção de soluções exatas, justificam o fato desta categoria de problemas ainda ser objeto de intensas pesquisas nas áreas da Pesquisa Operacional (PO) e da Inteligência Artificial (IA), uma vez que o estudo de tais problemas fornece uma base comum para análise e solução de outros problemas que pertencem à mesma categoria (LIU; MENG, 2008; ZHANG; DU, 2011; HOSEINI; SHAYESTEHE, 2010; GOLDBERG, 1989; DYCKHOFF, 1990; DORIGO; STÜTZLE, 2004).

Neste contexto, a computação bioinspirada, mais especificamente as técnicas meta-heurísticas bioinspiradas como os Algoritmos Genéticos e o Algoritmo Colônia de Formigas, vem sendo cada vez mais utilizadas na solução de tais problemas (CASTRO ; ZUBEN, 2005), (DORIGO; STÜTZLE, 2004), (GOLDBERG, 1989).

O Algoritmo Genético (AG) é uma abstração de modelos de evolução presentes na natureza que opera sobre uma população de indivíduos, por meio da aplicação de mecanismos de seleção, cruzamento e mutação, gerando novos indivíduos que, a cada geração, tornam-se mais aptos.

O Algoritmo Colônia de Formigas é inspirado no comportamento de colônias de formigas que são capazes de encontrar o caminho mais curto entre suas colônias e as fontes de

alimento, influenciadas pela presença de feromônios neste caminho, essas formigas tendem a seguir na direção em que a concentração de feromônios é mais forte.

Além de aplicadas individualmente, essas técnicas podem ser combinadas para obtenção de um sistema mais poderoso, com menos deficiências e sinérgico (GOLDSCHMIDT, 2005). Esse tipo de aplicação vem ganhando espaço na resolução dos mais diversos tipos de problemas (ZHANG; DU, 2011; SASSI, 2006; PINTO, 2011; MANTAWY et al., 1999; LIU; MENG, 2008).

## 1.1 JUSTIFICATIVA E MOTIVAÇÃO

A otimização do corte bidimensional guilhotinado está fundamentada na necessidade da indústria vidreira em minimizar o desperdício de material e, com isso, melhorar a sua competitividade no mercado vidreiro.

Tentativas de solução desse problema aplicando técnicas são bem-vindas, o que justifica a aplicação de meta-heurísticas bioinspiradas, mais especificamente o Algoritmo Genético e o Algoritmo Colônia de Formigas.

A motivação reside em tentar melhorar os resultados obtidos na solução do problema do corte bidimensional guilhotinado em uma indústria vidreira por meio da aplicação individual e da combinação das duas técnicas.

## 1.2 PROBLEMA DA PESQUISA

Resultados positivos podem ser obtidos na solução do problema do corte bidimensional guilhotinado em uma indústria vidreira quando o Algoritmo Genético e o Algoritmo Colônia de Formigas são aplicados individualmente e combinados?

## 1.3 OBJETIVO GERAL

Este trabalho tem como objetivo geral aplicar meta-heurísticas bioinspiradas ao problema do corte bidimensional guilhotinado em uma indústria vidreira.

### 1.3.1 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Implementar e aplicar, individualmente e combinadas, as meta-heurísticas bioinspiradas, Algoritmo Genético (AG) e Otimização por Colônia de Formigas ou ACO (do inglês *Ant Colony Optimization*), na solução do problema do corte bidimensional guilhotinado em uma indústria vidreira.

- Implementar uma variação das heurísticas de encaixe *First-fit* (FF) e *Best-first* (BF) para auxiliar na aplicação do AG e do ACO.

- Selecionar instâncias da literatura para aplicação das meta-heurísticas bioinspiradas.

- Selecionar softwares comerciais de otimização do corte bidimensional guilhotinado para validação dos Resultados (*Benchmarking*).

## 1.4 ORGANIZAÇÃO DO TRABALHO

Com a Introdução (Capítulo 1) este trabalho está estruturado em cinco capítulos:

Capítulo 2 - Fundamentação Teórica. Neste capítulo são apresentados os conceitos dos temas abordados no desenvolvimento do trabalho.

Capítulo 3 – Materiais e Métodos. Neste capítulo são apresentadas a metodologia de pesquisa, as ferramentas e o levantamento das informações utilizadas.

Capítulo 4 - Apresentação e Discussão dos Resultados. Neste capítulo são apresentados e discutidos os resultados.

Capítulo 5 – Conclusões e perspectivas futuras.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos dos temas abordados no desenvolvimento do trabalho. Iniciando-se por um breve histórico sobre a indústria vidreira seguido pela descrição do problema de corte e empacotamento, do problema de corte bidimensional, da otimização combinatória e dos problemas de satisfação de restrições e finalizando com a descrição de meta-heurísticas.

### 2.1 A INDÚSTRIA VIDREIRA

A indústria vidreira é dividida em diversos segmentos de acordo com o produto que fabrica. No Brasil, ela concentra suas atividades nos segmentos de vidros planos e domésticos. Os vidros planos são fabricados em chapas e são utilizados, principalmente, na construção civil, na indústria automobilística e decorações de interiores, principalmente espelhos e janelas. Os vidros domésticos são usados em utensílios como copos, louças, e outros objetos diversos de decoração.

O Quadro 1 mostra breve cronologia da indústria vidreira no Brasil (ASSIS; TEIXEIRA, 2001).

**Quadro 1 – Breve cronologia da indústria vidreira no Brasil**

<b>Período</b>	<b>Acontecimentos</b>
1882	Inauguração, no Rio de Janeiro, da primeira grande indústria brasileira de vidros, a Fábrica Esberard, produtora de vidros de embalagem e vidros planos, empregando mais de quinhentos funcionários.
1895	Inauguração da Companhia Vidraria Santa Marina, que se tornou outro empreendimento de grande sucesso, empregando seiscentos funcionários e fabricando, em menos de dez anos de existência, um milhão de garrafas e dois mil metros quadrados de vidro plano por mês.
1909	No censo das atividades econômicas realizadas, publicado pelo Centro Industrial do Brasil, o setor vidreiro aparece apenas em 29º lugar entre as 38 indústrias mais importantes, segundo a relação entre o valor da produção anual e o capital registrado.
1939 - 1945	A revolução industrial brasileira foi impulsionada pela Segunda Guerra Mundial, o que aumentou as dificuldades de abastecimento externo, estimulou os empreendimentos internos em busca da autossuficiência industrial e mudou o foco do mercado, que foi direcionado para os setores considerados básicos para o desenvolvimento das atividades industriais, tais como os fornecedores de insumos e os produtores de bens de consumo.
1959	Com o avanço industrial no setor automotivo brasileiro e o surgimento de um novo sistema de fabricação conhecido como <i>float</i> e a evolução dos processos de fabricação de vidros existentes, foi possível o desenvolvimento de uma variedade de tipos de vidros temperados, laminados, vidros refletivos para controle solar, vidros duplos para controle termo acústico e outras tantas evoluções.
1982	Inauguração da Cebrace I marcou o início de uma bem sucedida parceria entre a Pilkington e a Saint-Gobain (empresa que se firmou como a maior indústria vidreira francesa e um dos principais fabricantes de vidro da Europa) na atividade industrial de vidro plano no Brasil, por intermédio de suas filiais brasileiras, Blindex e Santa Marina, respectivamente.
1989	Inauguração da fábrica Cebrace II em Caçapava, resultante da parceria entre a Pilkington e a Saint-Gobain.
1996	Inauguração da fábrica Cebrace III em Jacareí com capacidade produtiva de 600 toneladas/dia, aumentando a capacidade produtiva diária da Cebrace a 1800 toneladas diárias, e tornando-a a maior fabricante brasileira e sul-americana de vidro plano.

Fonte: Assis e Teixeira (2001).

Segundo Assis e Teixeira (2001) a partir da década de 1950, o sistema *float* representou uma revolução tecnológica na história do vidro plano, levando a indústria do vidro plano a outro nível de desempenho técnico e econômico. Este sistema foi introduzido em 1959 pela Pilkington que, após quase dez anos de pesquisa e de experimentação, conseguiu, afinal, consolidar um novo sistema de produção de chapas de vidro que

possibilitou ao vidro plano uma qualidade muito superior àquela possibilitada pelos sistemas anteriores de produção.

Na Figura 1 pode-se ver a extensão da área onde seria inaugurada, em 1989, a fábrica da Cebrace II.

**Figura 1** – Vista panorâmica da Providro e da terraplenagem para a construção da Cebrace II, que seria inaugurada em 1989.



Fonte: Assis e Teixeira (2001).

O processo de fabricação do sistema *float* ocorre em 5 etapas (PEREIRA, 2006):

a) **Composição** – à mistura vitrificável é adicionado o vidro partido (caco) para diminuir a temperatura de fusão. O transporte, a pesagem, a mistura e enformamento são feitos automaticamente. Esta mistura é umedecida para evitar a segregação dos grãos das diferentes matérias-primas e a libertação de poeiras.

b) **Forno de fusão** - a elaboração do vidro compreende três fases essenciais: a fusão, durante a qual as matérias-primas são fundidas a temperaturas próximas dos 1150°C; a afinação, durante a qual o vidro fundido é tornado homogêneo e livre de bolhas gasosas; o acondicionamento térmico, onde o vidro pouco viscoso é arrefecido até que a sua viscosidade corresponda às exigências do processo de transformação.

c) **Banho de estanho** - o vidro líquido é vertido a cerca de 1000°C sobre um banho de estanho fundido. O vidro, menos denso que o estanho, “flutua” sobre este e forma uma chapa com uma espessura natural de 6 a 7 mm. As faces do vidro são polidas, por um

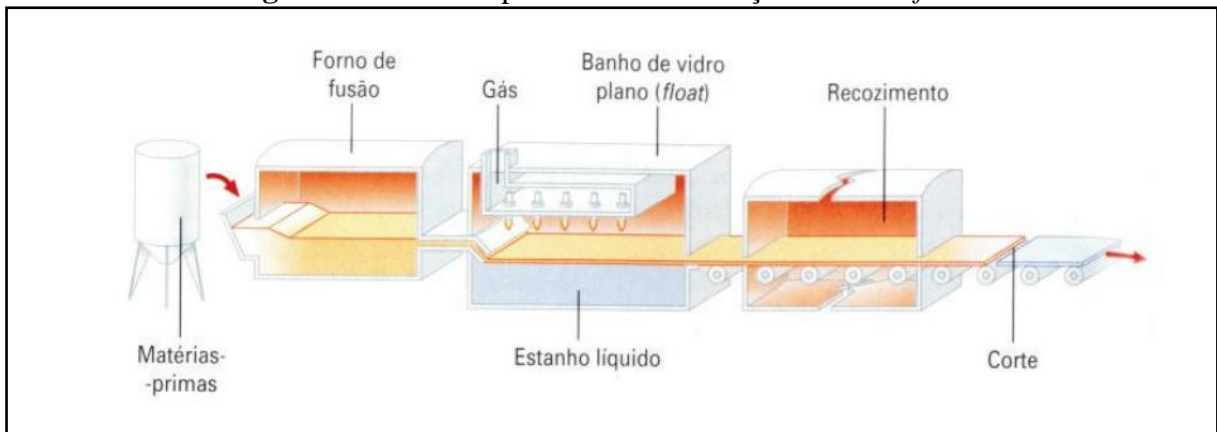
lado pela superfície do estanho, e pelo outro pelo fogo. Dispositivos mecânicos permitem acelerar ou diminuir a velocidade do vidro para regular a sua espessura.

d) **Forno de recozimento** - à saída do banho de estanho a chapa de vidro agora rígida, passa para o túnel de arrefecimento. A temperatura do vidro é reduzida gradualmente de 620 a 250 °C. O arrefecimento lento passa a ser realizado ao ar livre. Este processo permite libertar o vidro de todas as tensões internas que poderiam provocar a sua quebra no momento do corte.

e) **Corte** - a chapa de vidro frio, até aqui contínua, é cortada automaticamente em placas de dimensões padronizadas.

As fases do processo de fabricação do vidro *float* podem ser vistas na Figura 2.

**Figura 2** – Fases do processo de fabricação do vidro *float*.



Fonte: Pereira, 2006.

Podem-se verificar na Figura 2 as diferentes fases do processo utilizado no sistema *float*, desde a entrada da matéria-prima, passando pela fusão, o banho de estanho, o recozimento e, finalmente, o corte.

Antes da introdução do sistema *float*, a indústria vidreira utilizava o sistema soprado, no qual o vidro:

- Era aquecido até tornar-se flexível por meio de um forno especial.
- Eram colhidas gotas de vidro na extremidade de um tubo de metal.
- Mediante o sopro, essas gotas eram transformadas em cilindros.
- As bases do cilindro eram cortadas e o cilindro partido em dois.
- O cilindro era aquecido num forno ou inserido em uma câmara de estiramento onde era puxado verticalmente para se planificar.

## 2.2 PROBLEMAS DE ALOCAÇÃO DE RECURSOS

Os problemas de alocação de recursos em geral, têm sido um fator de grande impacto no desempenho dos sistemas de produção industrial e podem ser encontrados nos mais diversos processos industriais e de produção, tais como o corte de materiais (MORABITO; PUREZA, 2007; CANTO et al., 2010), empacotamento (PUREZA; MORABITO, 2003; SILVA; SOMA, 2003), escalonamento de mão de obra (CONSTANTINO et al., 2006), Escalonamento de tarefas (SILVA, 2006), localização de facilidades (PRADO, 2007), distribuição de bens e serviços (SHERAFAT, 2013), planejamento de produção (FREITAS; VIEIRA, 2010; PEREIRA; COSTA, 2012; PERGHER et al.; 2014).

O Quadro 2 mostra exemplos práticos de cada um desses processos.

**Quadro 2** – Processos industriais e de produção que envolvem alocação de recursos.

<b>Processo</b>	<b>Exemplo prático</b>
<b>Corte de Materiais</b>	Uma fábrica de vidro vende peças sob encomenda que são produzidas cortando-se placas grandes em pedaços menores. Uma placa grande pode ser cortada de diversas maneiras e sempre haverá um desperdício de vidro oriundo dos pedaços que sobram após o corte das peças desejadas. Esses pedaços não podem ser aproveitados para produzir peças úteis.
<b>Empacotamento</b>	Os problemas de empacotamento podem ser encarados como o inverso dos problemas de corte. A ideia é encontrar a melhor maneira de agrupar um conjunto de itens de modo que o espaço total necessário para guardá-los seja minimizado.
<b>Escalonamento de mão de obra</b>	Dado um conjunto de tarefas a realizar e um conjunto de funcionários, um empresário deseja encontrar a melhor maneira de alocar seus funcionários às tarefas de forma que todas as tarefas sejam cumpridas e os gastos com mão de obra sejam minimizados.
<b>Escalonamento de tarefas</b>	Em certas fábricas, um produto final é criado a partir da execução de pequenas tarefas. Essas tarefas possuem regras de precedência entre si e particularidades que exigem um ou outro tipo de máquina para sua execução. Com isso, dado um conjunto de itens a produzir, deseja-se descobrir, para cada máquina da fábrica, a ordem em que as tarefas devem ser processadas de forma a minimizar o tempo de produção.
<b>Localização de facilidades</b>	Dado um conjunto de clientes que precisam ser atendidos e um conjunto de possíveis locais para instalação de facilidades, deseja-se determinar quais os melhores locais para instalação das facilidades de forma que todos os clientes sejam atendidos a um custo mínimo.
<b>Distribuição de bens e serviços</b>	Dado um conjunto de clientes que precisam ser atendidos, deve-se fazer a roteirização dos veículos que serão utilizados, considerando-se a quantidade de carga a ser colocada em cada veículo e quais veículos irão atender quais clientes ou regiões.
<b>Planejamento de produção</b>	Dado um horizonte de demanda por produtos, um fabricante de determinado item de consumo precisa decidir quanto deve produzir por mês de forma a atender toda a demanda e ainda minimizar os custos.

Além dos problemas citados no Quadro 2, o problema do corte bidimensional (PCB) representa um dos principais problemas de alocação de recursos da indústria vidreira.



### 2.3 O PROBLEMA DO CORTE BIDIMENSIONAL GUILHOTINADO NA INDÚSTRIA VIDREIRA

O corte bidimensional guilhotinado é um problema de alocação de recursos complexo (TEMPONI, 2007), cujo tempo necessário para encontrar uma solução aumenta à medida que aumenta a quantidade de peças a serem cortadas de uma mesma chapa.

Com o aumento da complexidade, torna-se impraticável o cálculo da solução exata para esse problema, já que o tempo necessário para resolvê-lo torna-se inaceitável para os requerimentos da solução (SEDGWICK, 2002).

O corte bidimensional guilhotinado é um problema de otimização combinatória (que significa encontrar, dentre todos os possíveis subconjuntos, aquele cujo custo seja o menor possível) que consiste na determinação de padrões de corte de unidades de material, de maneira a produzir um conjunto de unidades menores, satisfazendo determinadas restrições.

Caprara e Toth (2001) introduziram a seguinte formulação matemática para o PCB (Equações 1 a 6):

Minimizar:

$$\sum_{k=1}^K Y_k \quad (1)$$

Sujeito a:

$$\sum_{k=1}^K X_{ik} = 1, \quad i = 1, \dots, n, \quad (2)$$

$$\sum_{i=1}^n W_i X_{ik} \leq A_{yk}, \quad k = 1, \dots, K, \quad (3)$$

$$\sum_{i=1}^n V_i X_{ik} \leq B_{yk}, \quad k = 1, \dots, K, \quad (4)$$

$$Y_k \in \{0,1\}, \quad k = 1, \dots, K, \quad (5)$$

$$X_{ik} \in \{0,1\}, \quad i = 1, \dots, n, k = 1, \dots, K, \quad (6)$$

Onde:

$K$  é o número máximo de objetos necessários;

$n$  é o número de itens;

$W_i$  é a medida do item na primeira dimensão;

$V_i$  é a medida do item na segunda dimensão;

$A$  e  $B$  são as medidas do objeto na primeira e segunda dimensão, respectivamente.

Cada variável  $Y_k = 1$  se o objeto  $k$  é usado ou  $Y_k = 0$ , caso contrário.

$X_{ik} = 1$  se o item  $i$  foi inserido no objeto  $k$  ou  $X_{ik} = 0$ , caso contrário.

A formulação matemática definida por Caprara e Toth (2001) pode ser considerada fraca, pois não especifica o número de cortes necessários e nem a posição e orientação dos itens.

A principal restrição do processo de corte bidimensional guilhotinado em uma indústria vidreira, como o próprio nome sugere, refere-se ao fato de que a disposição das peças a serem cortadas deve, necessariamente, possibilitar um corte “guilhotinável” e contínuo, maximizando, sempre que possível, a utilização do material envolvido no processo (COSTA; SASSI, 2012).

Segundo Temponi (2007), o Problema de Corte e Empacotamento (PCE) é um termo geral adotado para definir uma classe de problemas de otimização combinatória que consiste na combinação de unidades menores (itens) dentro de unidades maiores (objetos).

Devido aos Problemas de Corte (PC) e Empacotamento (PE) possuírem características idênticas, ambos podem ser descritos de forma similar (TEMPONI, 2007):

- no Problema de Corte (PC) o objetivo principal consiste em determinar a forma de cortar objetos para produzir os itens;

- no Problema de Empacotamento (PE), o objetivo consiste em encaixar (empacotar) os itens dentro dos objetos.

Em ambos os casos, os itens devem ser selecionados e agrupados em conjuntos que são atribuídos aos objetos de forma que respeitem algumas restrições (TEMPONI, 2007):

- o item de um conjunto não pode ser maior do que o objeto ao qual será atribuído;

- os itens de um conjunto devem caber inteiramente no objeto ao qual foram atribuídos;

- os itens devem ser dispostos nos objetos sem nenhuma sobreposição.

Segundo Wäscher et al. (2007) os PCE podem ser divididos em cinco subcategorias, cada uma com objetivos e restrições específicos:

- Problema de seleção de objetos:** quando determinados objetos possuem características (custo, dimensões, material) diferentes;

- Problema de seleção de itens:** quando algum item tiver prioridade em relação aos outros;

- Problema de agrupamento de itens:** quando um determinado conjunto de itens deve ficar separado de outros;

**-Problema de alocação de itens em objetos:** quando alguns itens precisarem ser alocados apenas em determinados objetos;

**-Problema de layout:** quando os itens tiverem que ser distribuídos nos objetos, respeitando-se alguma condição geométrica.

Dyckhoff (1990) classificou os PCE segundo quatro características principais:

**- Dimensão do problema**

- (1) unidimensional
- (2) bidimensional
- (3) tridimensional
- (N) N-dimensional ( $N > 3$ )

**- Forma de alocação das unidades**

- (V) seleção de unidades grandes
- (B) seleção de unidades pequenas

**- Sortimento de unidades grandes**

- (O) uma unidade
- (I) unidades de tamanhos iguais
- (D) unidades de tamanhos diferentes

**- Sortimento de unidades pequenas**

- (F) poucas unidades de tamanhos diferentes
- (M) muitas unidades de muitos tamanhos diferentes
- (R) muitas unidades de poucos tamanhos diferentes
- (C) unidades de tamanhos iguais

Embora tenha definido uma estrutura comum entre os PCE, a tipologia de Dyckhoff (1990) possui algumas desvantagens, apontadas por Wascher et al. (2007) como, por exemplo, no caso do corte unidimensional onde um grande número de itens com poucas formas diferentes é produzido de uma quantidade ilimitada de objetos, podendo-se incorrer em duas situações caso os objetos possuam diferentes tamanhos:

- na primeira situação, os objetos são separados em pequenos grupos de elementos semelhantes;

- na segunda situação, todos os objetos são diferentes.

Em ambas as situações, a classificação segundo a tipologia de Dyckhoff seria 1/V/D/R.

Os PCE podem ser classificados segundo Wäscher et al. (2007) quanto:

- à dimensionalidade: unidimensional, bidimensional, tridimensional.

- aos tipos de itens: idênticos, pouco heterogêneos, muito heterogêneos.
- aos tipos de objetos: um único objeto com todas as dimensões fixas ou com uma ou mais dimensões variáveis; muitos objetos idênticos, pouco heterogêneos ou muito heterogêneos.

- à forma dos itens: regulares (retangulares) ou irregulares (não retangulares).

Na indústria vidreira o PCE mais comum é o Problema de Corte Bidimensional (PCB), que, por sua vez, pode ser classificado como não guilhotinado ou guilhotinado.

No Problema de Corte Bidimensional Não Guilhotinado (PCBNG), o corte não possui a restrição de ter que se estender de um lado a outro do objeto e apenas acompanha o contorno do item até que o mesmo tenha sido totalmente destacado do objeto.

No caso do Problema de Corte Bidimensional Guilhotinado (PCBG), o corte deve se estender de um lado ao outro do objeto, de forma que, a cada corte, sejam produzidos dois novos retângulos (ou sub-objetos).

Segundo Lodi et al. (1999), conforme a sua tipologia, os problemas de corte bidimensional podem ser classificados em 4 tipos:

- 2BP|O|G: os itens são orientados (O), ou seja, não podem ser rotacionados, e o corte precisa ser guilhotinado (G);

- 2BP|R|G: os itens podem ser rotacionados em 90 graus (R) e é preciso que o corte seja guilhotinado;

- 2BP|O|F: os itens são orientados e o corte é livre (F), ou seja, não precisa ser guilhotinado.

- 2BP|R|F: os itens podem ser rotacionados em 90 graus e o corte é livre.

Este trabalho irá abordar apenas o problema do tipo 2BP|R|G (os itens podem ser rotacionados e o corte deverá ser guilhotinado sem limitação do número de estágios), segundo a classificação definida por Lodi et al. (1999), ou 2/V/I/M (bidimensional com seleção dos objetos e itens, com objetos semelhantes e itens muito diferentes) segundo a tipologia de Dyckhoff (1990).

O PCBG ocasiona uma das principais dificuldades observadas na indústria vidreira, que se refere à determinação do custo final dos produtos vendidos. Esta dificuldade se dá pelo fato de os produtos não seguirem uma padronização de tamanho ou formato.

Na indústria vidreira, cada ordem de corte (pedido com as medidas das peças requisitadas pelos clientes que precisarão ser cortadas a partir de uma chapa maior) contém características específicas para atender necessidades específicas de cada cliente, e nem sempre

é possível determinar se as sobras decorrentes do corte de uma determinada ordem poderão ser utilizadas como base para outras ordens de corte.

Neste contexto, a indústria vidreira tem duas possibilidades:

-Considerar que as sobras serão utilizadas para cortar peças de outras ordens de corte, deduzir o seu custo do preço do produto final e correr um risco ter prejuízo caso isso não ocorra.

-Considerar que as sobras não serão utilizadas para cortar peças de outras ordens de corte, acrescentar o seu custo no preço do produto final e correr o risco de perder competitividade comercial.

Baseado em tais fatos, as soluções para o PCBG são de grande importância para a indústria vidreira no que se refere à existência de um processo de otimização do corte, onde, a partir de vários pedidos, se determina a melhor sequência e disposição do corte, de forma a maximizar a utilização de material e, conseqüentemente, conseguir mensurar o custo do produto final de uma forma mais confiável.

O PCBG é um problema de otimização combinatória, complexo e com restrições, no qual a complexidade para se determinar uma solução aumenta à medida que aumenta a quantidade de itens a serem cortados. Com o aumento da complexidade, torna-se impraticável o cálculo da solução exata (determinística) para esse problema, já que o tempo necessário para resolvê-lo torna-se inaceitável para os requerimentos da solução.

Estas características enquadram o PCBG em diversas classes de problemas de otimização combinatória tidos como intratáveis, que, segundo Sedgewick (2002), são aqueles para os quais não se conhece um algoritmo exato que garanta a solução em um espaço de tempo razoável, os chamados Problemas de Satisfação de Restrições (PSR).

No que se refere aos problemas tidos como intratáveis, para Sedgewick (2002), apesar desta classe de problemas possuir características que permitam utilizar métodos de força bruta (métodos que permitem tentar todas as possibilidades de combinações possíveis) para resolvê-los, eles são considerados intratáveis porque as possibilidades a considerar para a sua solução são muitas.

## 2.4 OTIMIZAÇÃO COMBINATÓRIA E PROBLEMAS DE SATISFAÇÃO DE RESTRIÇÕES

O mesmo problema pode ser tratado de diversas maneiras e, frequentemente, pode ser resolvido com inúmeras técnicas que diferem em eficiência e complexidade. As diferenças entre as técnicas podem não ser relevantes no processamento de uma pequena quantidade de dados (instâncias pequenas), porém se tornam mais relevantes proporcionalmente à quantidade de dados.

São exemplos clássicos de problemas de otimização combinatória: o problema do caixeiro viajante (LAWLER et al., 1985), o problema da mochila (MARTELLO; TOTH, 1990), o problema das  $n$  rainhas (BERNARD, 1990), o problema de corte unidimensional (GILMORE; GOMORY 1961; GILMORE; GOMORY, 1963; DYCKHOFF, 1981; WASCHER; GAU, 1996), o problema de corte bidimensional (MORABITO et al. , 1992; HOPPER; TURTON, 1999).

Para se comparar a eficiência das técnicas que se propõem a resolver problemas de otimização combinatória, utiliza-se uma medida de dificuldade de algoritmos conhecida como complexidade computacional, que, segundo Drozdek (2008), indica quanto esforço é necessário para se aplicar um algoritmo, ou quão custoso ele é. Este custo pode ser medido de diversas maneiras, mas as principais delas se referem aos critérios de eficiência de tempo e espaço.

As técnicas para resolução dos problemas de otimização combinatória, normalmente, tem o objetivo de maximizar ou minimizar uma função  $f$  em um domínio  $D$ , geralmente com um conjunto de restrições  $(R_1, R_2, R_3, \dots, R_n)$ . Por este motivo, geralmente os problemas de otimização combinatória também se enquadram na classe dos Problemas de Satisfação de Restrições (PSR).

Um PSR consiste de três componentes,  $V$ ,  $D$  e  $R$  (RUSSEL, 2010):

- $V$  é um conjunto de variáveis,  $\{V_1, \dots, V_n\}$ .
- $D$  é um conjunto de domínios  $\{D_1, \dots, D_n\}$ , um para cada variável.
- $R$  é um conjunto de restrições que especifica as combinações de valores permitidas.

No qual:

- Cada domínio  $D_i$ , consiste de um conjunto de valores permitidos,  $\{v_1, \dots, v_k\}$ , para a variável  $V_i$ .

- Cada constante  $C_i$  consiste de um par  $(e, r)$  no qual  $e$  é uma sequencia de variáveis que fazem parte da restrição e  $r$  é a relação que define os valores que estas variáveis podem assumir.

Uma relação  $r$  pode ser representada como uma lista explícita de todas as sequencias de valores que satisfaçam a restrição.

Uma atribuição que não viole nenhuma das restrições é chamada de atribuição consistente ou válida. Uma atribuição completa possui todas as variáveis mencionadas, e uma solução para um PSR é uma atribuição completa que satisfaz a todas as restrições. Uma atribuição parcial é a que atribui valores para apenas algumas variáveis (RUSSELL, 2010).

Basicamente, existem dois tipos de restrições, aquelas que precisam ser satisfeitas, pois implicam diretamente na obtenção da solução e aquelas cujo resultado positivo é desejável, mas não obrigatório para a solução do problema (LINDEN, 2012).

A busca de soluções para os PSR normalmente envolve:

- A formulação matemática do problema, que consiste na representação dos seus domínios, variáveis envolvidas no problema e das suas respectivas restrições.

- A busca de uma solução aceitável para o problema, que satisfaça todas as restrições, ou a prova de que não existe uma solução aceitável para o problema (DORIGO; STÜTZLE, 2004).

Quanto o maior o número de variáveis envolvidas no PSR mais complexo se torna encontrar uma solução ótima para ele, principalmente quando o método de otimização não é suficientemente poderoso na execução do processo de busca da solução. Nestes casos, mesmo que seu domínio seja mensurável, torna-se inviável em termos práticos e de complexidade computacional testar todas as combinações, mesmo em domínios de tamanho finito e moderado.

As técnicas de otimização abordadas na literatura (LIU; MENG, 2008; ZHANG; DU, 2011; HOSEINI; SHAYESTEH, 2010; GOLDBERG, 1989; DYCKHOFF, 1990; DORIGO; STÜTZLE, 2004) que trata de problemas de otimização combinatória envolvem uma série de termos e conceitos necessários para o seu entendimento.

A seguir são listados alguns desses termos relacionados com os problemas de otimização em geral:

**-Espaço de busca:** É o domínio, conjunto, espaço ou região que compreende as soluções possíveis ou viáveis sobre as variáveis do projeto do problema a ser otimizado, sendo delimitado pelas funções de restrição.

**-Função Objetivo:** É a função de uma ou mais variáveis de projeto que se quer otimizar, minimizando-a ou maximizando-a.

**-Máximo e Mínimo local:** Diz-se que um valor  $f(a)$  é um máximo local (ou relativo) de  $f$  se existe um intervalo aberto  $(c, d)$  contendo  $a$ , tal que  $f(x) \leq f(a)$ , para todos os valores de  $x$  em  $(c, d)$ . Analogamente, diz-se que um valor  $f(b)$  é um mínimo local ou relativo de  $f$  se existe um intervalo aberto  $(c, d)$  contendo  $b$ , tal que  $f(b) \leq f(x)$ , para todos os valores de  $x$  em  $(c, d)$ .

**-Máximo e Mínimo Global:** O máximo global (ou absoluto) de uma função é definido como o maior valor da função considerando todos os pontos do seu domínio. Já o mínimo global (ou absoluto) de uma função é definido como o menor valor da função considerando todos os pontos do seu domínio.

**-Ponto Ótimo:** É o ponto formado pelas variáveis de projeto que extremizam a função objetivo e satisfazem as restrições.

**-Restrições:** são funções de igualdade ou desigualdade sobre as variáveis de projeto que descrevem situações de projeto consideradas não desejáveis.

**-Valor Ótimo:** É o valor da função objetivo no ponto ótimo.

**-Variáveis de projeto:** são as que se alteram durante o processo de otimização, podendo ser contínuas ou discretas.

**- Lower Bound e Upper Bound:** São valores ótimos teóricos para um determinado problema de otimização combinatória.

As técnicas para se resolver problemas de otimização combinatória geralmente são baseadas em métodos determinísticos ou não determinísticos (probabilísticos). As técnicas de otimização combinatória baseadas nos métodos determinísticos geram uma sequência determinística de todas as soluções possíveis.

Nos métodos determinísticos, tanto a função objetivo quanto as restrições são dadas como funções matemáticas. Além disso, segundo Bastos (2004) a função objetivo deve ser contínua e diferenciável no espaço de busca.

As técnicas de otimização baseadas em métodos não determinísticos usam apenas a avaliação da função objetivo e introduzem dados e parâmetros aleatórios no processo de otimização.

As principais vantagens dos métodos não determinísticos em relação aos determinísticos são:



- a função objetivo e as restrições não precisam necessariamente ter uma representação matemática;
- não requerem que a função objetivo seja contínua ou diferenciável;
- trabalham adequadamente, tanto com parâmetros contínuos quanto com discretos, ou ainda com uma combinação deles;
- não necessitam de formulações complexas ou reformulações para o problema;
- não há restrição alguma quanto ao ponto de partida dentro do espaço de busca da solução;
- Otimizam um grande número de variáveis, desde que a avaliação da função objetivo não tenha um custo computacional demasiadamente alto.

A maior vantagem dos métodos não determinísticos em relação aos métodos determinísticos é o menor tempo de processamento.

Dentre as técnicas baseadas nos métodos não determinísticos, destacam-se as técnicas classificadas como meta-heurísticas: Algoritmos Genéticos (FALKENAUER, 1998; LINDEN, 2012), Algoritmo de Otimização Colônia de Formigas (DORIGO; STÜTZLE, 2004; CARVALHO, 2007; GUANGDONG; PING; QUN, 2007), o *Simulated Annealing* (HOSEINI; SHAYESTEH, 2010) e a *Particle Swarm Optimization* (KENNEDY; EBERHART, 1995).

## 2.5 HEURÍSTICAS E META-HEURÍSTICAS

Segundo definição de Zanakis e Evans (1981) as heurísticas são algoritmos que apresentam bons resultados ou soluções factíveis de modo rápido e fácil para vários problemas semelhantes, porém que não existem evidências de que apresentem soluções eficientes para todos os tipos de problemas.

Dentre as técnicas heurísticas mais abordadas na literatura (XAVIER; MIYAZAWA, 2007; MARTELLO; TOTH, 1990; FALKENAUER, 1998; MANTAWY; ABDEL-MAGID; SELIM, 1999) estão: Heurísticas de busca local e Heurísticas construtivas (*First-fit*, *Best-fit*, *Next-fit*, *Worst-fit*).

Segundo Miyazawa (2008), a busca local começa com uma atribuição ou solução e iterativamente fazem operações locais melhorando a solução anterior, devolvendo a solução anterior quando não for possível melhorar.

As Heurísticas Construtivas utilizam técnicas de adição na construção da solução do problema. A cada iteração vão sendo agregados pontos aos resultados parciais. Esta construção é um processo contínuo e gradativo (SCHOPF et al., 2004).

No caso dos problemas de corte unidimensional e bidimensional, as heurísticas construtivas mais utilizadas são as seguintes:

- *Next-Fit* (NF): cada item é inserido sequencialmente após o anterior e, caso não haja espaço suficiente para acomodá-lo no objeto corrente, um novo objeto é inserido e o item é acomodado neste objeto. O processo se repete até que todos os itens tenham sido acomodados, sendo que o espaço disponível nos objetos anteriores não é mais utilizado na acomodação dos itens subsequentes, ocasionando em um desperdício de espaço maior.

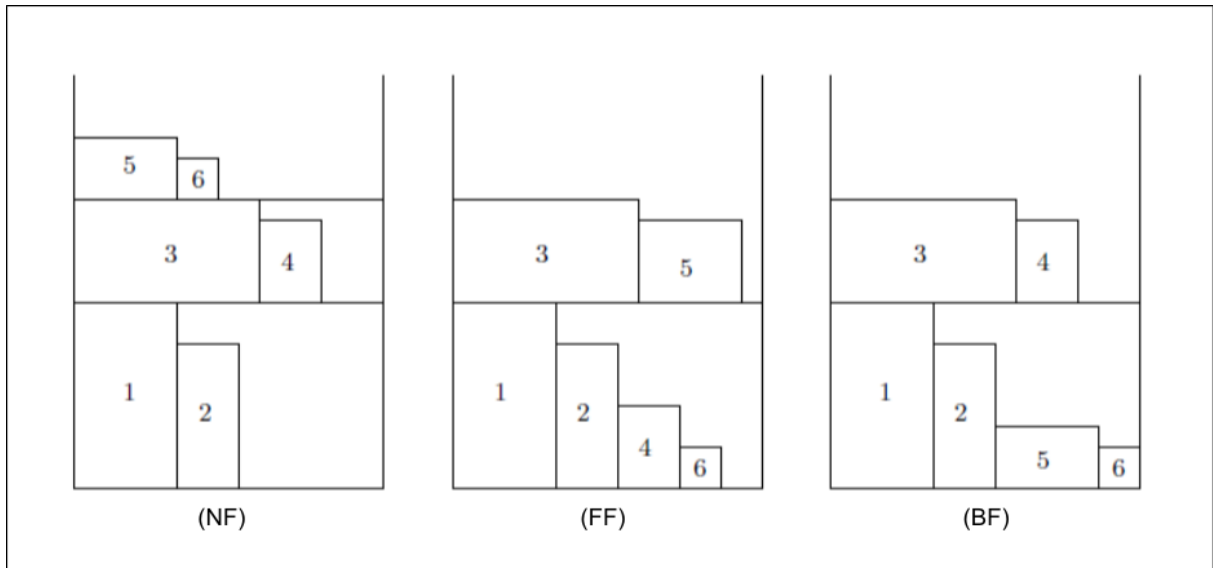
- *First-fit* (FF): os itens são inseridos no primeiro objetos com espaço suficiente para acomodá-los e, caso não haja nenhum objeto com espaço suficiente para acomodar o item, um novo objeto é inserido e o item é acomodado neste objeto. O processo se repete até que todos os itens tenham sido acomodados.

- *Best-fit* (BF): semelhante à FF, porém, ao invés de acomodar cada item no primeiro objeto disponível, existe uma busca em todos os objetos de forma a identificar o objeto que melhor acomodará o item. A heurística BF normalmente apresenta melhores resultados que a FF, porém com um maior custo computacional.

- *Worst-fit* (WF): normalmente é usada em problemas de alocação de memória e não é aplicável nos problemas de corte.

Na Figura 3 pode-se observar a diferenças entre as heurísticas *Next-Fit*, *First-fit* e *Best-fit*.

**Figura 3** – Itens encaixados a partir da heurística *Next-fit* (NF), *First-fit* (FF) e *Best-fit* (BF).



Fonte: Adaptado de Lodi et al. (1999)

Segundo Dorigo e Stuzze (2004) meta-heurística é um conjunto de conceitos algorítmicos que podem ser usados para definir métodos heurísticos aplicáveis a um conjunto extenso de problemas, ou seja, são heurísticas com utilidade mais generalista que podem ser aplicadas a diversas classes de problemas.

A utilização de meta-heurísticas tem incrementado significativamente a habilidade de encontrar soluções de qualidade e em um espaço de tempo aceitável para problemas relevantes e difíceis que envolvem otimização combinatória (LINDEN, 2012).

## 2.6 COMPUTAÇÃO BIOINSPIRADA

A Computação biologicamente inspirada, ou bioinspirada, é o campo de investigação que recorre a metáforas ou modelos teóricos dos sistemas biológicos, a fim de projetar ferramentas computacionais ou sistemas para resolver problemas complexos. Os resultados alcançados são algoritmos ou sistemas que têm uma semelhança (por vezes superficial) com os fenômenos ou modelos biológicos estudados (CASTRO; ZUBEN, 2005).

O Algoritmo Genético e o Algoritmo Colônia de Formigas são técnicas meta-heurísticas bioinspiradas da Inteligência Artificial que, devido a sua eficiência na busca de soluções em espaços de busca muito grandes, vem sendo cada vez mais aplicadas na solução de problemas complexos, principalmente problemas de otimização combinatória.

As técnicas meta-heurísticas bioinspiradas, mais especificamente o Algoritmo Genético e o Algoritmo Colônia de Formigas, têm se mostrado particularmente úteis na solução de problemas complexos que envolvem otimização combinatória (LINDEN, 2012; DORIGO; GAMBARDELLA, 1997; HOPPER; TURTON, 1999).

### 2.6.1 Algoritmos Genéticos

O princípio de funcionamento das técnicas chamadas evolutivas ou evolucionárias que tem como um dos representantes os Algoritmos Genéticos (AG) baseia-se na interpretação das teorias de Darwin (Teoria da Evolução) como um processo adaptativo de otimização.

Dado um problema específico, as técnicas evolutivas tentam encontrar uma solução a partir da criação e manipulação de uma população de estruturas candidatas à solução do problema. Cada indivíduo da população é submetido a processos de seleção baseados em medidas de adaptabilidade e a operadores genéticos (cruzamento e mutação) durante vários ciclos denominados gerações. Juntos, esses mecanismos visam criar uma população de indivíduos de alta adaptabilidade, ou seja, indivíduos que tenham alta probabilidade de ser a solução procurada do problema.

A partir da década de sessenta, surgiram diversos paradigmas computacionais inspirados em modelos simplificados da Teoria da Evolução. Entre eles o Algoritmo Genético (AG), proposto por Holland (1975) e que influenciou outros dois paradigmas conhecidos como: Sistemas Classificadores (SCs) (BELEW; FORREST, 1988) e Programação Genética (PG) (KOZA, 1992).

O Algoritmo Genético (AG) foi primeiramente proposto por John Holland nos anos 60 e desenvolvido por ele e seus estudantes durante os anos 60 e 70 na Universidade de Michigan (HOLLAND, 1975). Holland apresentou o AG como uma abstração de modelos de evolução presentes na natureza.

O Algoritmo Genético é uma meta-heurística de otimização, que opera sobre uma população de indivíduos (estruturas de dados que representam candidatos a solução de um problema) por meio da aplicação de mecanismos de seleção, cruzamento (*crossover*) e mutação, gerando novos indivíduos que, a cada geração, tornam-se mais aptos e, portanto, mais próximos da solução ótima para o problema.

Tem sido aplicado em diversos tipos de problemas, tais como o problema do caixeiro viajante (LINDEN, 2012), problemas de corte (COSTA et al., 2010), processamento de imagens (HOSEINI; SHAYESTEH, 2010), dentre outros.

A terminologia utilizada pelos AGs consiste basicamente de analogias oriundas da Biologia real, embora as entidades a que se referem sejam muito mais simples do que as entidades reais biológicas (MITCHELL, 1996). Abaixo seguem os termos mais utilizados:

**Cromossomo:** Todos os organismos vivos são constituídos por células e cada célula contém o mesmo conjunto de um ou mais cromossomos. Os cromossomos descrevem o organismo.

**Genoma:** É a coleção completa de todo o material genético de um organismo (todos os cromossomos juntos).

**Genótipo:** Se refere ao conjunto particular de genes contidos em um genoma. Dois indivíduos que possuam genomas idênticos são referidos como possuindo o mesmo genótipo.

**Gene:** Divisão conceitual de um cromossomo. Grosso modo, pode-se dizer que um gene codifica uma característica, como a cor dos olhos, por exemplo.

**Locus:** Posição em que o gene se localiza no cromossomo.

**Alelo:** Característica ou valor numérico que representa um gene. Por exemplo, um cromossomo que utiliza codificação binária, possui em cada posição (locus) dois possíveis alelos: 0 e 1.

**População:** Conjunto dos cromossomos que compõe cada geração.

**Diploide:** População cujos cromossomos são formados em pares.

**Haploide:** População cujos cromossomos não estão dispostos em pares.

**Cruzamento (*crossover*):** Troca de partes entre dois cromossomos, geralmente haploides.

**Gameta:** Cada cromossomo gerado por cruzamento.

**Aptidão (*Fitness*):** Probabilidade que um organismo possui para reproduzir (viabilidade), ou uma função do número de descendentes que este organismo possui (fertilidade).

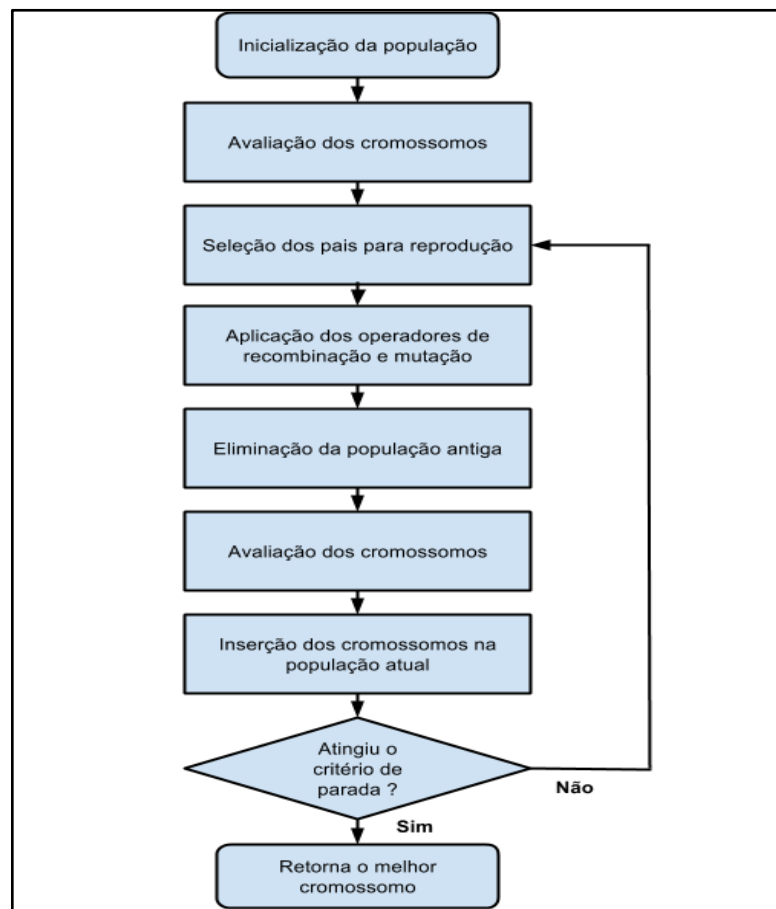
Segundo Mitchell (1996), nos AGs, o termo cromossomo se refere a uma solução candidata para o problema tratado e geralmente as aplicações de AGs utilizam indivíduos haploides (indivíduos de cromossomos únicos).

O comportamento padrão de um AG pode ser resumido, algoritmicamente pelos seguintes passos e o seu fluxograma pode ser conferido na Figura 4 (LINDEN, 2012):

- a) Inicialização da população de cromossomos.
- b) Avaliação de cada cromossomo da população.
- c) Seleção dos pais que serão cruzados para gerar novos cromossomos.

- d) Aplicação dos operadores de recombinação e mutação aos pais selecionados, gerando os indivíduos da nova geração.
- e) Eliminar os membros da população antiga (anterior).
- f) Avaliar todos os novos cromossomos e inseri-los na população.
- g) Se atingir o critério de parada, retorna o melhor cromossomo, caso contrário, voltar para o passo c.

**Figura 4** – Fluxograma com a sequência padrão de execução de um AG.



Fonte: o autor.

Para resolver problemas de otimização combinatória por meio do AG, tem-se que entender como as seguintes questões devem ser resolvidas (LINDEN, 2012):

**a)** Qual a representação adotada para os cromossomos?

A representação dos cromossomos é fundamental para o AG, de forma que, quanto mais adequada ela for ao problema, maior a qualidade dos resultados obtidos.

**b)** Para esta representação, qual é o mecanismo dos operadores genéticos?

A escolha correta dos operadores é de extrema importância para o AG. O operador de mutação, por exemplo, é fundamental para um AG, garantindo a diversidade da população, enquanto que o operador de cruzamento contribui para a igualdade entre os cromossomos e o operador de elitismo garante que os melhores cromossomos não se perderão entre as gerações.

**c) Qual a função de avaliação utilizada?**

É através da função de avaliação que os indivíduos mais adaptados são selecionados e submetidos aos operadores genéticos, dando origem a uma nova população. Assim, é fundamental que a sua definição leve em conta as características do problema a ser tratado, de forma a possibilitar uma avaliação eficaz dos indivíduos de cada geração e, conseqüentemente, melhorando a qualidade dos resultados obtidos.

A seguir, são descritos os principais passos utilizados na implementação de um AG:

**a) Codificação dos Cromossomos**

A representação dos cromossomos consiste, basicamente, em traduzir as informações do problema a ser tratado em uma maneira viável a ser tratada pelo computador (LINDEN, 2012).

Ao aplicar um AG na resolução de problemas que envolvem otimização combinatória, é fundamental que a sua representação seja adequada ao tipo de problema que será tratado. No problema do caixeiro viajante (CARVALHO, 2007), por exemplo, o importante é a ordem em que as cidades serão visitadas. Já nos problemas de corte e empacotamento, além da ordem em que os itens serão combinados para melhor aproveitamento dos objetos, é importante a relação entre os itens semelhantes, ou seja, o agrupamento entre os itens de forma a conseguir um melhor aproveitamento dos objetos (DYCKHOFF, 1990), (FALKENAUER, 1998), (HOPPER; TURTON, 1999).

Assim, no caso do problema de corte bidimensional guilhotinado é importante utilizar uma representação que contenha todos os itens que serão cortados, colocados em uma ordem específica, o que levará a optar pela representação em lista de todos os elementos presentes no problema. Em um problema onde seria necessário otimizar a disposição de corte de 10 itens (peças de vidro), por exemplo, a representação baseada em ordem de uma das soluções poderia ser a seguinte: 10, 8, 1, 2, 4, 5, 6, 7, 9, 3.

Apesar da codificação baseada em ordem atender ao critério relacionado com a ordem das peças a serem cortadas no problema de corte bidimensional guilhotinado, ela deixa de lado outro fator desta mesma classe de problemas. Este fator refere-se ao agrupamento entre os itens, já que os itens a serem cortados formam um agrupamento nos objetos de onde serão cortados (FALKENAUER, 1998).

### **b) Seleção**

O operador de seleção do AG é uma analogia com a seleção biológica natural de Darwin. Realizar uma seleção significa escolher, dentre os indivíduos pertencentes a uma população, aqueles que serão recombinados, de forma a criar descendentes para a próxima geração.

O conceito fundamental é o de privilegiar os indivíduos com função de avaliação alta, sem desprezar completamente os indivíduos com função de avaliação baixa.

Esta decisão deve-se ao fato de que um indivíduo, apesar de ter uma avaliação péssima, pode possuir características genéticas favoráveis à geração de um indivíduo melhor.

Se fossem utilizados apenas os melhores indivíduos no processo de reprodução, isso poderia afetar a diversidade das gerações posteriores e, conseqüentemente, impedir que a evolução ocorra de forma satisfatória.

### **c) Cruzamento (*Crossover*)**

O Cruzamento consiste no operador de acasalamento, que permite a produção de novos indivíduos através da troca de informações parciais entre pares de cromossomos.

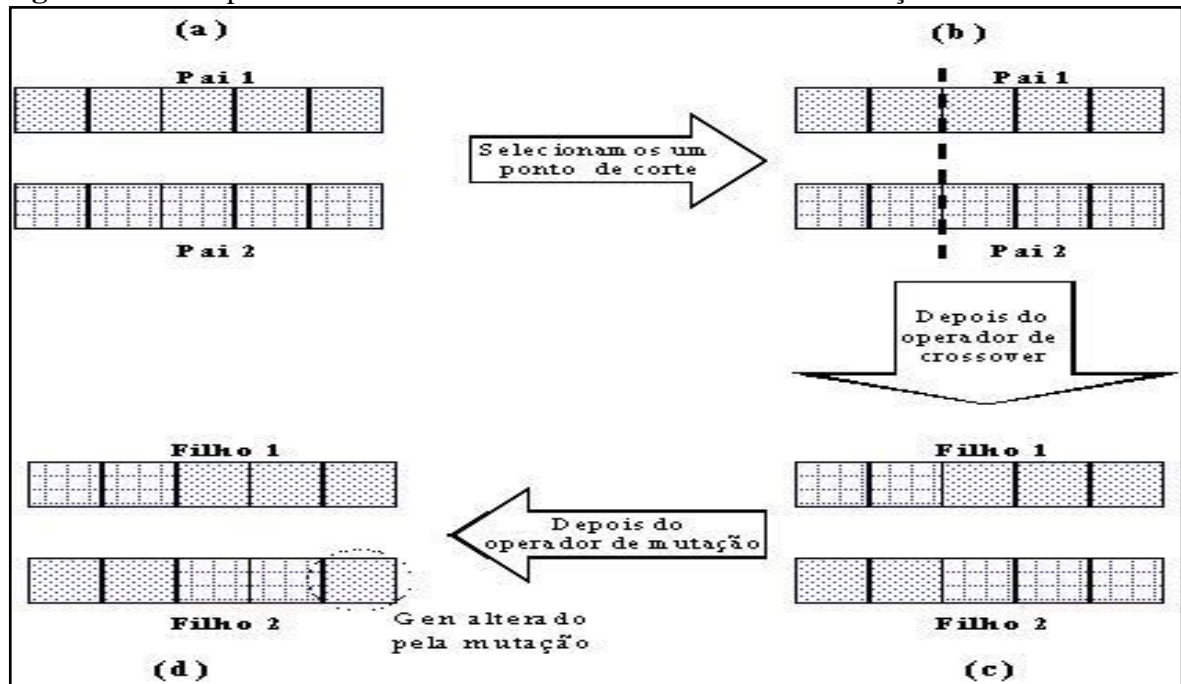
Depois de selecionados os pais pelo operador de seleção, são definidos aleatoriamente os pontos de corte (uma posição entre dois genes que compõem o material genético de cada cromossomo). Normalmente cada indivíduo de  $n$  genes pode conter até  $n-1$  pontos de corte.

No caso dos cruzamentos com um ponto de corte, após a definição deste, o primeiro filho é composto por meio da concatenação da parte do primeiro pai à esquerda do ponto de corte com a parte do segundo pai à direita do ponto de corte. O segundo filho é composto por meio da concatenação das partes que sobraram (a metade do segundo pai à esquerda do ponto de corte com a metade do primeiro pai à direita do ponto de corte).

O processo de cruzamento com um ponto de corte, entre dois cromossomos com codificação binária pode ser conferido na Figura 5. No passo (a) são selecionados os pais, no passo (b), definido o ponto de corte e nos passos subsequentes (c e d) são formados os filhos a partir de cada parte separada dos pais por meio dos pontos de corte.



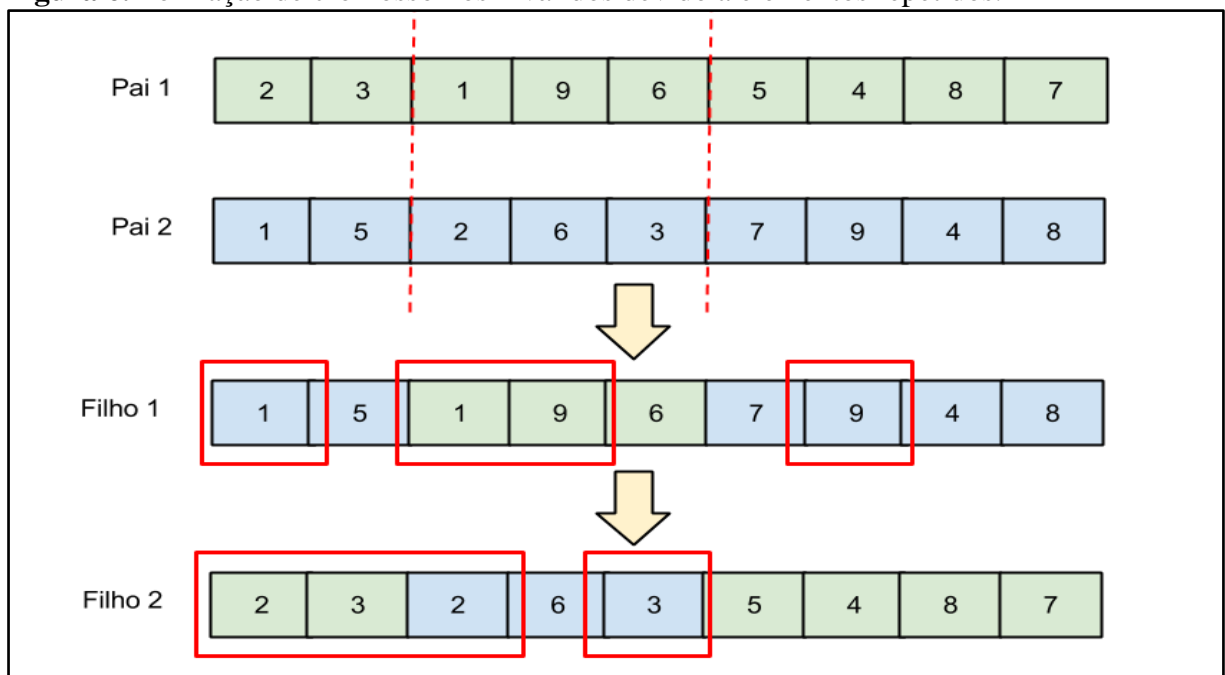
**Figura 5** – Exemplo de cruzamento entre cromossomos com codificação baseada em ordem.



Fonte: Linden (2012).

A aplicação do operador de *crossover* entre dois cromossomos com codificação baseada em ordem é um pouco mais complexa, pois se os filhos forem apenas formados a partir de cada parte separada dos pais por meio dos pontos de corte, poderiam ser gerados filhos com elementos repetidos, como pode ser conferido na Figura 6.

**Figura 6.** Formação de cromossomos inválidos devido a elementos repetidos.



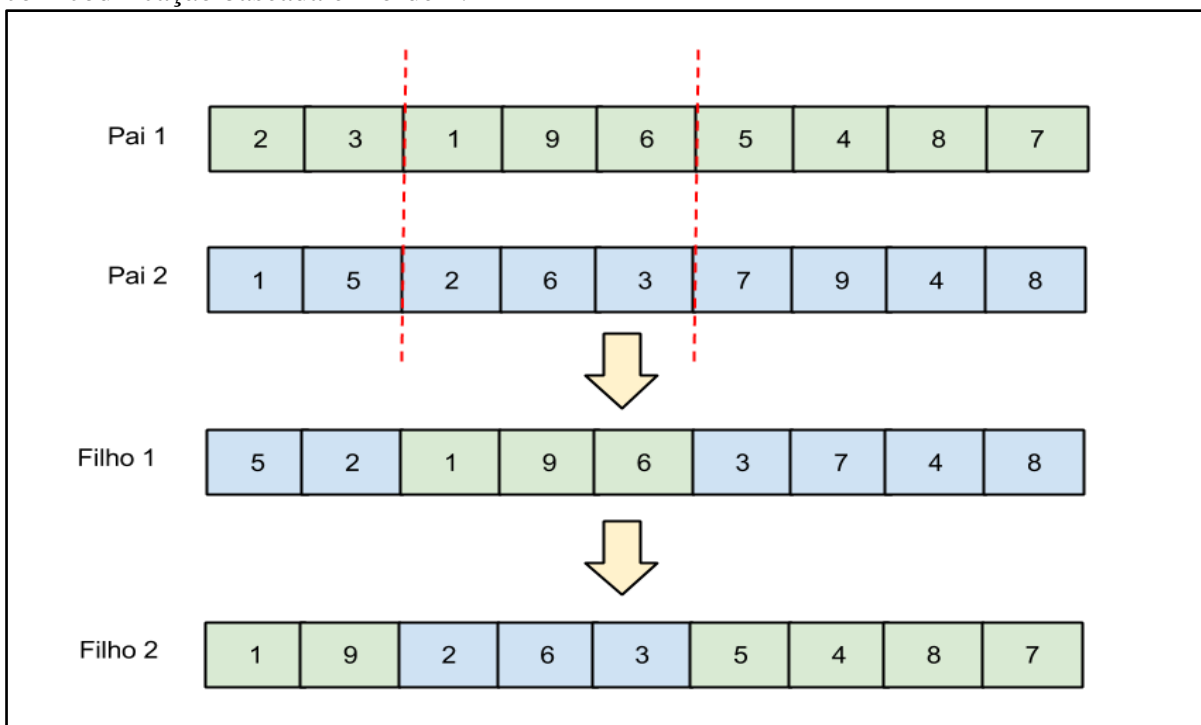
Fonte: o autor.

Para evitar a formação de filhos inválidos, na codificação baseada em ordem, o processo de *crossover* desses cromossomos deve preservar a ordem relativa dos genes ao invés de sua posição absoluta, como ocorre no caso da codificação binária. Esse processo de *crossover* dos cromossomos com codificação baseada em ordem, com dois pontos de corte, ocorre conforme os seguintes passos (LINDEN, 2012):

- selecionam-se os pontos de corte;
- copiam-se para o filho 1 os elementos (genes) do pai 1 entre os pontos de corte;
- forma-se uma lista com os elementos do pai 1 fora dos pontos de corte;
- permuta-se esta lista de forma que os elementos apareçam na mesma ordem que no pai 2;
- coloca-se estes elementos nos espaços do filho 1, na ordem gerada no passo anterior;
- repete-se o processo para gerar o filho 2, substituindo-se o pai 1 pelo pai 2 e vice-versa.

Um resultado da aplicação do operador de *crossover* de dois pontos dos cromossomos com codificação baseada em ordem pode ser visto na Figura 7.

**Figura 7.** Exemplo da aplicação do operador de *crossover* de dois pontos para cromossomos com codificação baseada em ordem.



Fonte: o autor.

#### **d) Mutação**

O operador de mutação consiste em alterar aleatoriamente e a partir de um fator probabilístico baixo alguma característica genética dos indivíduos. Para aplicação da mutação, sorteia-se um número aleatório entre 0 e 1, se for menor que a probabilidade determinada, então o operador atua sobre o gene em questão.

#### **e) Elitismo**

Elitismo é uma pequena modificação na geração da população que não impacta consideravelmente no tempo de processamento, e que garante que o desempenho do AG será crescente no decorrer das gerações. A ideia básica do elitismo é selecionar os  $n$  melhores indivíduos de cada geração e passar para a próxima visando garantir que suas características sejam preservadas.

#### **f) Função de Avaliação (*Fitness*)**

O objetivo da função de avaliação (*fitness*) é de medir o grau de aptidão (ou qualidade) de um indivíduo em relação ao restante da população, ou seja, a função de avaliação refere-se ao grau com que uma solução candidata contribui para a convergência do algoritmo na busca da melhor solução.

Segundo Linden (2012), a função de avaliação é como uma nota dada ao indivíduo na resolução do problema e, dada a generalidade dos AGs, em muitos casos ela é a única ligação verdadeira do programa (algoritmo) com o problema real. Isto se deve ao fato de que a função de avaliação só julga a qualidade da solução que está sendo apresentada por aquele indivíduo, sem armazenar qualquer tipo de informação sobre as técnicas de resolução do problema.

### 2.6.2 Algoritmo de Otimização por Colônia de Formigas

As formigas quando em busca de alimento exploram aleatoriamente o ambiente em torno da colônia de uma forma aparentemente desordenada. Ao percorrerem esse trajeto, as formigas liberam uma substância química chamada feromônio. Influenciadas pela presença de feromônios no caminho, as formigas tendem a seguir na direção em que a concentração de feromônios é mais forte.

Os experimentos com formigas reais demonstraram que essa coordenação entre formigas via trilhas de feromônio produz um comportamento coletivo de auto-organização, onde os caminhos mais curtos entre os seus ninhos e as fontes de alimento são

progressivamente seguidos pelas demais formigas ao reforçarem as trilhas de feromônio nas melhores rotas e, eventualmente, encontrarem o caminho mais curto. A Figura 8 mostra a tendência das formigas em seguir o caminho mais curto entre o ninho e a fonte de alimentos, em função da concentração de feromônios.

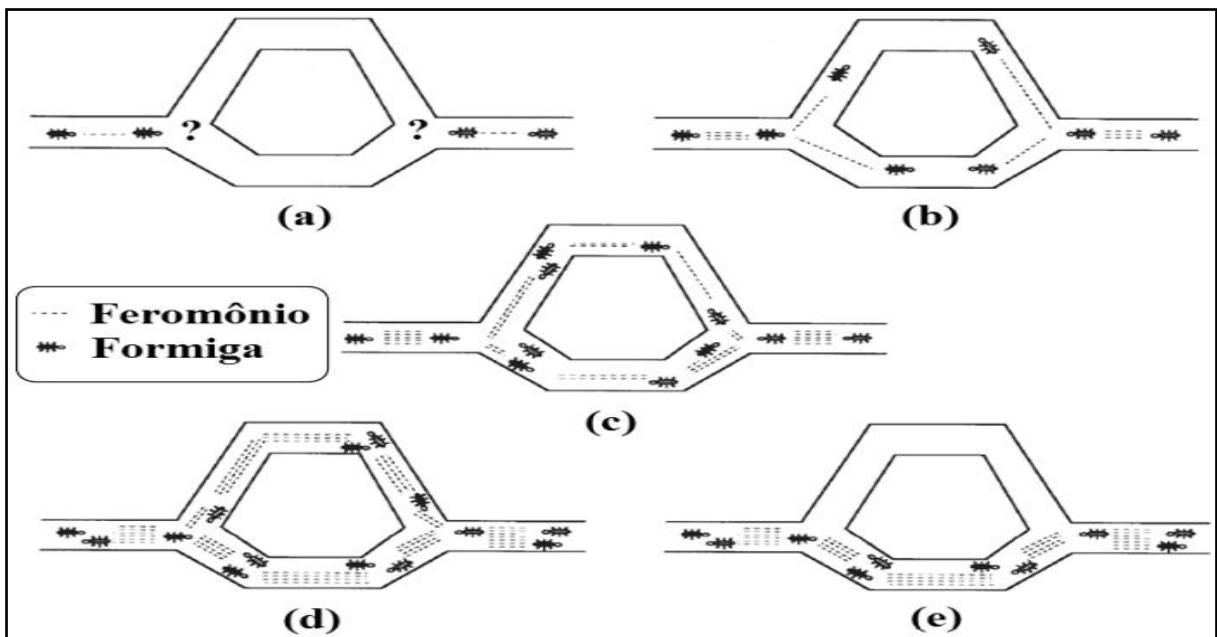
O Algoritmo Colônia de Formigas (ACO, do inglês *Ant Colony Optimization*) é uma meta-heurística bioinspirada, na qual formigas artificiais cooperam entre si, de forma a encontrar soluções ótimas para problemas difíceis de otimização combinatória.

O primeiro Algoritmo baseado no comportamento das colônias de formigas é conhecido como *Ant System* (AS) e foi apresentado pela primeira vez por Colomni et al. (1991) aplicado na resolução de um problema do Caixeiro Viajante.

Além do problema do Caixeiro Viajante, o ACO tem sido aplicado na resolução de problemas diversos, tais como o processamento de imagens (HOSEINI; SHAYESTE, 2010), problemas de carregamento (ZHANG; DU, 2011), problemas de corte (LEVINNE; DUCATELLE, 2003), dentre outros.

Como ilustra a Figura 8 abaixo, as formigas começam a percorrer os caminhos entre o ninho e a fonte de alimento aleatoriamente (a). À medida que percorrem os caminhos, liberam feromônios (b). A concentração de feromônios é maior nos caminhos por onde passam mais formigas (c). Assim, as formigas tendem a seguir o trajeto mais curto, ou seja, com a maior concentração de feromônios (d) e (e).

**Figura 8** – Tendência das formigas em seguir o trajeto mais curto em função da concentração de feromônios.



Fonte: Adaptado de Dorigo e Gambardella (1997).

O ACO pode ser representado como a ação combinada de três procedimentos (DORIGO; STÜTZLE, 2004, p.37):

- **Construção das soluções:** uma colônia de formigas é gerenciada de forma concorrente e assíncrona, construindo soluções incrementais para um determinado problema de otimização, a partir do uso das trilhas de feromônio e da informação heurística (que pode ser, por exemplo, a distância entre cidades adjacentes, no caso do problema do caixeiro viajante).

Cada formiga que constrói uma solução, ou enquanto uma solução é construída, avalia a solução (parcial) de forma a possibilitar que o procedimento seguinte decida a quantidade de feromônios a depositar.

- **Atualização de Feromônios:** é o procedimento a partir do qual as trilhas de feromônio são modificadas de forma incremental (quando as formigas depositam feromônio nas conexões usadas) ou decremental (devido à evaporação dos feromônios nas conexões menos utilizadas).

Do ponto de vista prático, o depósito de novos feromônios aumenta a probabilidade de uma conexão ser utilizada por um número maior de formigas, ou que seja utilizada por pelo menos uma formiga que produzirá uma boa solução, a qual será usada novamente por futuras formigas. Já a evaporação é uma forma bastante útil de esquecimento, evitando uma convergência prematura do algoritmo para uma região sub-ótima, favorecendo, assim, a exploração de novas áreas no espaço de busca.

- **Processamento complementar:** é utilizado para implementar ações centralizadas que não podem ser realizadas pelas formigas normais, como, por exemplo, a ativação de um procedimento de busca local.

A forma como estes três procedimentos irão interagir na implementação do ACO fica a cargo da natureza do problema ao qual ele será aplicado. Além do problema do caixeiro viajante, o ACO tem sido aplicado em diversos outros problemas de otimização combinatória, tais como roteamento de veículos (REIMANN et al., 2002), colorização de grafos (COSTA; HERTZ, 1997) e problemas de corte (LEVINNE; DUCATELLE, 2003).

Além de serem aplicadas individualmente para solução de problemas de otimização combinatória, as meta-heurísticas bioinspiradas AG e ACO têm sido frequentemente combinadas em busca de soluções melhores (GUANGDONG et al., 2007; HOSEINI; SHAYESTEH, 2010; CARVALHO, 2007).

Segundo Goldschmidt (2005), técnicas podem ser combinadas para gerar as chamadas arquiteturas híbridas ou sistemas híbridos. A grande vantagem desse tipo de sistema deve-se

ao sinergismo obtido pela combinação de duas ou mais técnicas. Este sinergismo reflete na obtenção de um sistema mais poderoso (em termos de interpretação, de aprendizado, de estimativa de parâmetros, de generalização, dentre outros) e com menos deficiências.

Existem vários trabalhos que combinam técnicas formando uma arquitetura híbrida (CARVALHO, 2007; SASSI, 2006; AFFONSO, 2010; KAUPA, 2013). Três são as formas básicas de se associarem duas técnicas para a construção de uma arquitetura híbrida (SOUZA, 1999):

**-Híbrida Sequencial:** nesta forma, uma técnica atua como entrada de outra técnica.

**-Híbrida Auxiliar:** esta forma poderia ser exemplificada do seguinte modo: uma rede neural artificial invoca um AG para a otimização de seus pesos ou de sua estrutura. Neste caso, tem-se um maior grau de hibridização em comparação com o híbrido sequencial.

**-Híbrida Incorporada:** nesta forma praticamente não há separação entre as duas técnicas. Pode-se dizer que a primeira técnica possui a segunda técnica e vice-versa. Poderia ser exemplificado por um sistema *Neuro-Fuzzy* híbrido em que um sistema de inferência *fuzzy* é implementado segundo a estrutura de uma rede neural artificial. Aqui a hibridização é a maior possível.

Os Materiais e Métodos utilizados neste trabalho são apresentados no próximo capítulo.

### 3 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os Materiais e Métodos utilizados no trabalho e a descrição detalhada da metodologia de realização dos experimentos.

#### 3.1 CARACTERIZAÇÃO METODOLÓGICA

A metodologia de pesquisa adotada neste trabalho foi definida como bibliográfica, exploratória e experimental. A pesquisa bibliográfica abrange a leitura, análise e interpretação de livros, periódicos, documentos, mapas, imagens, manuscritos, etc. Todo material recolhido foi submetido a uma triagem, a partir da qual foi possível estabelecer um plano de leitura. Tratou-se de uma leitura atenta e sistemática que se fez acompanhar de anotações e fichamentos que, eventualmente, serviram à fundamentação teórica do estudo (GIL, 2002).

A realização da pesquisa bibliográfica foi embasada em consultas a fontes bibliográficas e de referencial teórico, tais como: artigos, livros, teses, dissertações, sites com conteúdos sobre a indústria de vidro, problemas de corte na indústria vidreira, otimização, meta-heurísticas bioinspiradas e arquiteturas híbridas.

Foram consultadas as seguintes bases de dados: SCIELO, IEEE Xplore, SCOPUS e de congressos da área da Engenharia de Produção, como o Encontro Nacional de Engenharia de Produção (ENEGEP), o Simpósio de Engenharia de Produção (SIMPEP), o Simpósio Brasileiro de Pesquisa Operacional (SBPO) e o Encontro Mineiro de Engenharia de Produção (EMEPRO).

Segundo Yin (2006) a pesquisa exploratória permite uma maior familiaridade entre o pesquisador e o tema pesquisado, visto que este ainda é pouco conhecido, pouco explorado. Nesse sentido, caso o problema proposto não apresente aspectos que permitam a visualização dos procedimentos a serem adotados, será necessário que o pesquisador inicie um processo de sondagem, com vistas a aprimorar ideias, descobrir intuições e, posteriormente, construir hipóteses.

Por ser uma pesquisa bastante específica, pode-se afirmar que ela assume a forma de estudo de caso, sempre em consonância com outras fontes que darão base ao assunto abordado, como é o caso da pesquisa bibliográfica (EISENHARDT, 1989).

Uma pesquisa exploratória visa proporcionar maior familiaridade com o problema com vistas a torná-lo explícito ou a construir hipóteses. Pode-se dizer que esta pesquisa objetiva o aprimoramento de ideias ou a descoberta de intuições. Seu planejamento é,

portanto, bastante flexível para que possibilite a consideração dos mais variados aspectos relativos ao fato estudado.

A pesquisa experimental determina um objeto de estudo, selecionam-se as variáveis que seriam capazes de influenciá-lo, definem-se as formas de controle e de observação dos efeitos que a variável produz no objeto (GIL, 2002).

### 3.2 FERRAMENTAS, PLATAFORMA DE ENSAIO E BASE DE DADOS

Para a realização dos experimentos (Fases 1 a 5 e Fase 7, descritas na seção 3.4,) foi utilizada uma base de dados do mundo real com 50 pedidos de corte fornecidos por uma indústria vidreira da cidade de São Paulo, contendo de 32 a 1056 peças com três atributos:

- largura das peças a serem cortadas;
- altura das peças a serem cortadas;
- o tamanho das chapas a serem utilizadas para o corte das peças.

Para a realização dos experimentos (Fase 6, descrita na seção 3.4) foram utilizadas instâncias tratadas por Lodi et al. (1999). Estas instâncias são divididas em 10 classes de tamanhos variados de itens e objetos.

A validação dos resultados (Fase 7, descrita na seção 3.4) foi realizada usando como *Benchmarking* os seguintes *softwares* comerciais de otimização do corte bidimensional: Opty Way e Lisec GPS.opt.

A plataforma de hardware utilizada nos experimentos foi um computador com processador Intel® Core™i5 de 1,7 GHz com 4,00 GB de memória RAM DDR3, 128 GB de disco rígido e sistema operacional MAC OS X 10.7.5 de 64 bits.

Para implementação das heurísticas e meta-heurísticas foi utilizada a linguagem de programação Java (Oracle) a partir do ambiente de desenvolvimento Eclipse (*Apache Software Foundation*) na sua versão 4.2.2. Toda a programação na linguagem Java foi realizada pelo autor deste trabalho.

### 3.3 METODOLOGIA EXPERIMENTAL

A Metodologia Experimental foi dividida em sete fases:

- **Fase 1:** Preparatória;
- **Fase 2:** Implementação e aplicação do AG;
- **Fase 3:** Implementação e aplicação do ACO;



- **Fase 4:** Implementação e aplicação das meta-heurísticas combinadas;
- **Fase 5:** Ajustes nas implementações e alterações na combinação das técnicas;
- **Fase 6:** Aplicação das meta-heurísticas em instâncias da literatura;
- **Fase 7:** Validação dos Resultados (*Benchmarking* com *softwares* comerciais de otimização do corte bidimensional).

Segue a descrição das fases que compõem a metodologia experimental:

#### - FASE 1: PREPARATÓRIA

Nesta fase foram selecionados, a partir da base de dados com 50 pedidos, 4 pedidos de corte chamados “Problemas”, utilizados em todas as fases da Metodologia Experimental.

Os 4 problemas escolhidos foram denominados de P1, P2, P3 e P4 (Tabela 1) e a sua escolha foi em função da quantidade média de itens dos pedidos da base de dados.

Para o processamento dos problemas P1, P2, P3 e P4 foram utilizados objetos de 6,0 x 3,21 metros, que se referem ao tamanho de chapa padrão utilizado nos pedidos de corte que constam na base de dados de 50 pedidos utilizada.

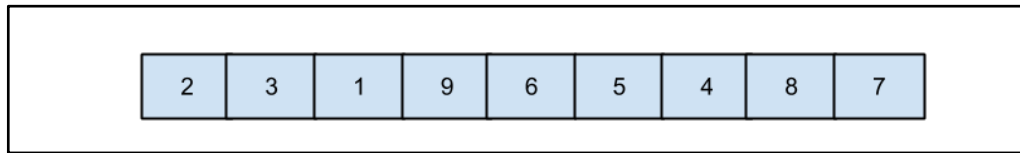
Para auxiliar na aplicação do AG e do ACO ao PCBG (Problema de Corte Bidimensional Guilhotinado) foi necessária a implementação de uma variação das heurísticas de encaixe *First-fit* (FF) e *Best-fit* (BF) (LODI et al., 1999).

Estas heurísticas, denominadas FFDWDH (*First-fit decreasing width decreasing height*) e BFDWDH (*Best-fit decreasing width decreasing height*) consistem nas heurísticas FF e BF originais com uma ordenação prévia dos itens, primeiro pela largura e em seguida pela altura. Esta ordenação tem o objetivo de permitir que os itens maiores sejam tratados primeiro e, ao mesmo tempo, diminuir a fragmentação nos resultados finais, já que os itens menores são inseridos por último nas áreas livres restantes.

A utilização dessas heurísticas como auxiliares do AG e do ACO se justifica pelo fato de as meta-heurísticas serem algoritmos genéricos que podem ser aplicados na resolução de diversas classes de problemas (DORIGO; STUZZLE, 2004) e que, para que possam apresentar resultados mais eficazes, geralmente precisam ser combinadas com técnicas mais específicas e direcionadas ao problema a ser tratado.

As primeiras populações (indivíduos ou formigas) são gerados aleatoriamente pelas meta-heurísticas. Cada solução representa uma sequência, a partir da qual, os itens serão encaixados ou alocados nos objetos. Um exemplo de indivíduo de uma população pode ser visto na Figura9.

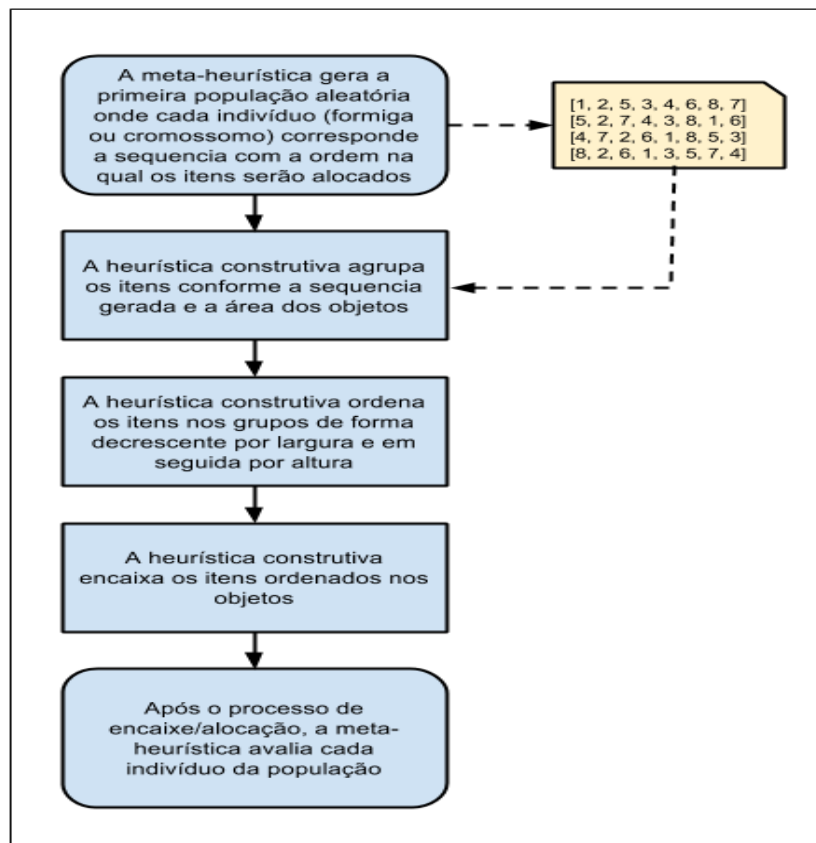
**Figura 9** – Exemplo de sequencia (solução, indivíduo ou formiga) gerada pela meta-heurística.



Fonte: O Autor.

A formação das soluções iniciais e a iteração entre as meta-heurísticas e heurísticas ocorre conforme o fluxograma da Figura 10.

**Figura 10** – Iteração entre as meta-heurísticas e heurísticas construtivas.



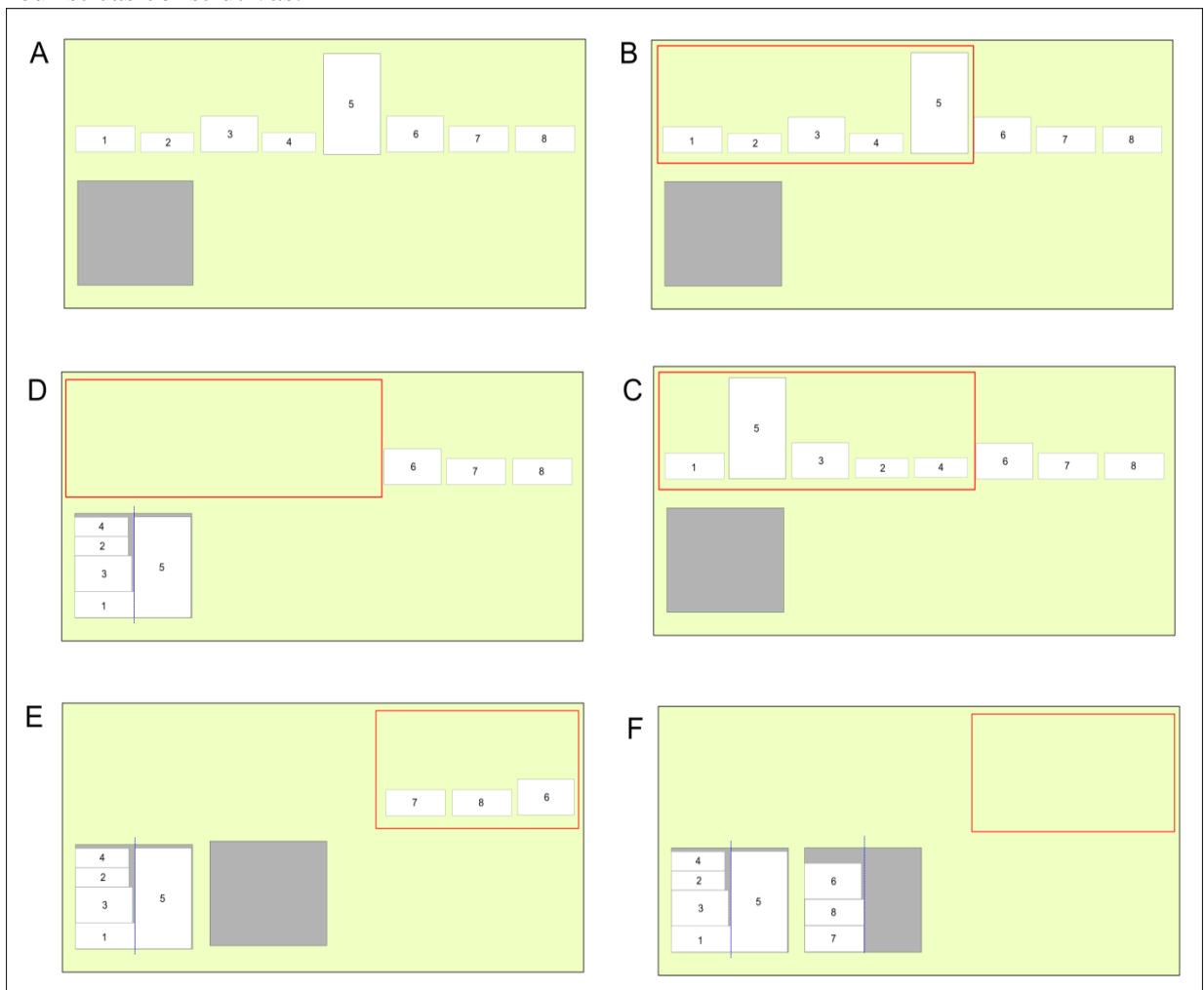
Fonte: O Autor.

A alocação ou encaixe dos itens nos objetos é feita pela heurística construtiva selecionada, conforme as sequências recebidas das meta-heurísticas, como pode ser visto na Figura 11, abaixo:

- Os itens são dispostos conforme sequência gerada pela meta-heurística (A);
- São selecionados os itens na sequência de forma que a soma de suas áreas seja menor ou igual à área do objeto (B);

- O subconjunto dos itens selecionados é ordenado de forma decrescente por largura e em seguida por altura (C);
- Os itens do subconjunto são sequencialmente encaixados no objeto, sendo que o primeiro item inserido gera uma coluna ou sub-objeto. Os itens seguintes que não puderem ser alocados nesta coluna, gerarão a próxima coluna, e assim sucessivamente, até que todos os objetos do subconjunto tenham sido encaixados (D);
- Quando o objeto atual não comportar mais itens, cria-se mais um objeto vazio e inicia-se o processo conforme descrito a partir do passo A, até que todos os itens tenham sido alocados (E, F).

**Figura 11** – Processo de alocação ou encaixe dos itens nos objetos, através da aplicação das heurísticas construtivas.



Fonte: o Autor.

Verificam-se na Tabela 1 as informações dos problemas P1, P2, P3 e P4 selecionados.

**Tabela 1** – Informações dos problemas P1, P2, P3 e P4.

Problema	Tamanho da chapa matriz	Quantidade de peças para cortar
P1	3210 x 6000	32
P2	3210 x 6000	70
P3	3210 x 6000	351
P4	3210 x 6000	1056

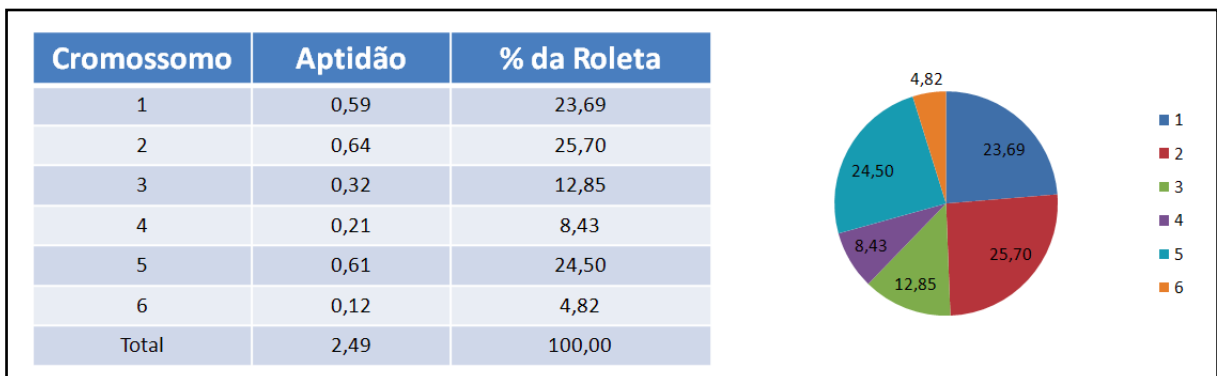
Fonte: Base de dados utilizada.

### - FASE 2: IMPLEMENTAÇÃO E APLICAÇÃO DO AG

O AG foi adaptado a partir de uma implementação de Linden (2012) modificando a função de avaliação, de forma a adequá-lo para resolução dos problemas de corte bidimensional guilhotinado. Esta implementação utilizada como base, pode ser aplicada na resolução de várias classes de problemas, alterando apenas a função de avaliação, que segundo Linden (2012), em muitos casos é a única ligação verdadeira do programa com o problema real.

A seleção dos indivíduos do AG para realização do operador de cruzamento foi feita utilizando o método da Roleta Viciada (LINDEN, 2012). Neste método indivíduos com alta aptidão recebem uma proporção maior na roleta e os indivíduos com menor aptidão uma porção relativamente menor da mesma.

Na Figura 12 estão dispostos seis pais com as respectivas aptidões (áreas dos setores circulares), para serem selecionados através da roleta e gerarem uma nova população. Observa-se que o indivíduo de número dois tem maiores chances de ser escolhido ao rodar a roleta.

**Figura 12** – Exemplo de aplicação da roleta viciada

Fonte: Adaptado de Linden (2012).

Foi utilizada a função objetivo representada na Equação 7, sugerida por Falkenauer (1992):

$$f_{BPP} = \frac{\sum_{i=1..N}(F_i/C)^k}{N} \quad (7)$$

Na qual:

$N$  é o número de objetos utilizados na solução.

$F_i$  é a soma das áreas dos itens no Objeto  $i$ .

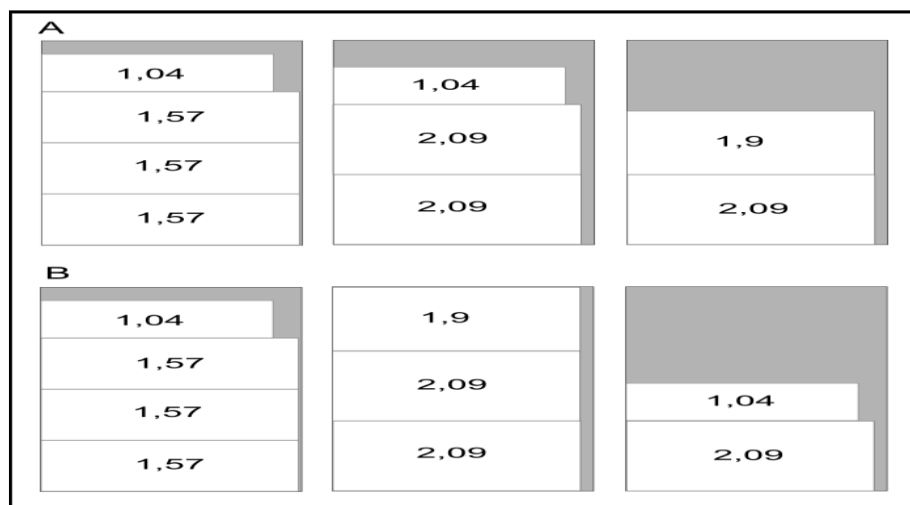
$C$  representa a capacidade, em metros quadrados, de cada Objeto.

$k$  é uma constante ( $k > 1$ ) que representa a concentração dos objetos melhor aproveitados em termos de espaço, em relação aos menos aproveitados (com maior desperdício de espaço).

A Figura 13 mostra um exemplo de aplicação desta função objetivo para diferenciar duas soluções iguais em relação a quantidade de objetos utilizados. Com  $N = 3$ ,  $k = 2$  e  $C = 6,4$  (metros quadrados), o resultado da função é 0,60 (solução A) e 0,64 (solução B), demonstrando que, apesar de utilizar a mesma quantidade de objetos, a solução B é melhor por apresentar um melhor aproveitamento do último objeto.

O valor de  $k=2$  foi sugerido por Falkenauer (1992) que, após realização de experimentos com diversos valores de  $k$ , concluiu que valores muito altos levam a uma convergência prematura do AG (ótimos locais), obtendo os melhores resultados com  $k=2$ .

**Figura 13-** Comparação de duas soluções a partir da função objetivo da Equação 7.



Fonte: o Autor.

Para calibração do AG, ele foi aplicado na resolução de cada problema (P1, P2, P2 e P4), variando-se os seguintes operadores (LINDEN, 2012):

- Quantidade de gerações.
- Percentual de mutação.
- Quantidade de cromossomos.
- Percentual de elitismo.
- Pontos de corte.

A cada alteração nos operadores, o AG foi aplicado por 5 vezes consecutivas em cada problema, totalizando 20 execuções, de forma a identificar o impacto da alteração individual de cada operador no resultado apresentado pelo AG.

### - FASE 3: IMPLEMENTAÇÃO E APLICAÇÃO DO ACO

A implementação do ACO foi baseada na implementação de Levine e Ducatelle (2003). O processo de calibração do ACO foi executado em função dos seguintes parâmetros (DORIGO; STÜTZLE, 2004):

- Quantidade de formigas.
- Número de iterações.
- Alfa (influência do feromônio).
- Beta (influência da informação heurística).

O parâmetro Alfa ( $\alpha$ ) se refere ao grau de influência da matriz de feromônios na construção das soluções, que por sua vez, se refere à escolha da próxima peça a ser cortada da chapa principal. Já o parâmetro Beta ( $\beta$ ) está relacionado com o grau de influência da informação heurística (que neste caso, representa a semelhança entre as peças e a possibilidade de serem cortadas da mesma chapa) na construção das soluções.

Na construção das soluções, a probabilidade de uma formiga  $k$ , escolher um determinado item  $j$  para ser alocado do objeto corrente  $b$ , a partir do último item alocado  $g$ , é dada pela Equação 8:

$$p_k(s, b, j) = \begin{cases} \frac{[\tau_b(j)]^\alpha \cdot [\eta(j)]^\beta}{\sum_{g \in J_k(s, b)} [\tau_b(g)] \cdot [\eta(g)]^\beta} & \text{se } j \in J_k(s, b) \\ 0 & \text{caso contrário} \end{cases} \quad (8)$$

Na qual:

$J_k(s, b)$  é o conjunto de itens que estão qualificados para inclusão no objeto corrente.

$\eta(j)$  é a área do item  $j$ .

$g$  é o último item alocado do objeto corrente  $b$ .

O valor de feromônio de um item  $j$  em um objeto  $b$  é dado pela Equação 9 e é igual a soma de todos os valores da matriz de feromônios entre os itens  $i$  e  $j$  que já estão no objeto  $b$ :

$$\tau_b(j) = \begin{cases} \frac{\sum_{i \in b} \tau(i,j)}{|b|} & \text{se } b \neq \{\} \\ 1 & \text{caso contrário} \end{cases} \quad (9)$$

Caso o objeto corrente  $b$  esteja vazio,  $\tau_b(j)$  é igual a 1.

Após a construção das soluções da iteração corrente, a atualização da matriz de feromônios é atualizada conforme a Equação 10, adaptada da versão do ACO conhecida como MAX-MIN *Ant System* (MMAS) de Stutzle e Hoos (2000). Esta versão do ACO foi escolhida por ser simples de implementar e, ao mesmo tempo, apresentar boa performance (LEVINE; DUCATELLE, 2003). Nela, apenas a melhor formiga (solução) de cada iteração atualiza a matriz de feromônios.

$\tau(i, j)$  é incrementado a cada vez que  $i$  e  $j$  estão juntos na mesma chapa, conforme a Equação 10:

$$\tau(i, j) = \rho \cdot \tau(i, j) + m \cdot f(s^{best}) \quad (10)$$

Na qual:

$\rho$  é um fator de evaporação do feromônio, utilizou-se o valor de 0,98 sugerido para o MMAS (DORIGO; STÜTZLE, 2004).

$m$  representa quantas vezes  $i$  e  $j$  aparecem juntas na melhor solução de cada iteração  $s^{best}$ .

Na avaliação da qualidade das soluções do ACO foi utilizada a mesma função objetivo da Fase 2 (Equação 7).

#### - FASE 4: IMPLEMENTAÇÃO E APLICAÇÃO DAS METAHEURÍSTICAS COMBINADAS

A aplicação das duas meta-heurísticas combinadas tem como objetivo utilizar as suas respectivas capacidades de encontrar soluções ótimas em grandes espaços de busca e minimizar a convergência para mínimos locais, também presente em ambas as técnicas.

Para implementação das meta-heurísticas combinadas, foram utilizadas as implementações descritas nas Fases 2 e 3, apenas com adição da parte que faz a combinação cooperativa entre elas.

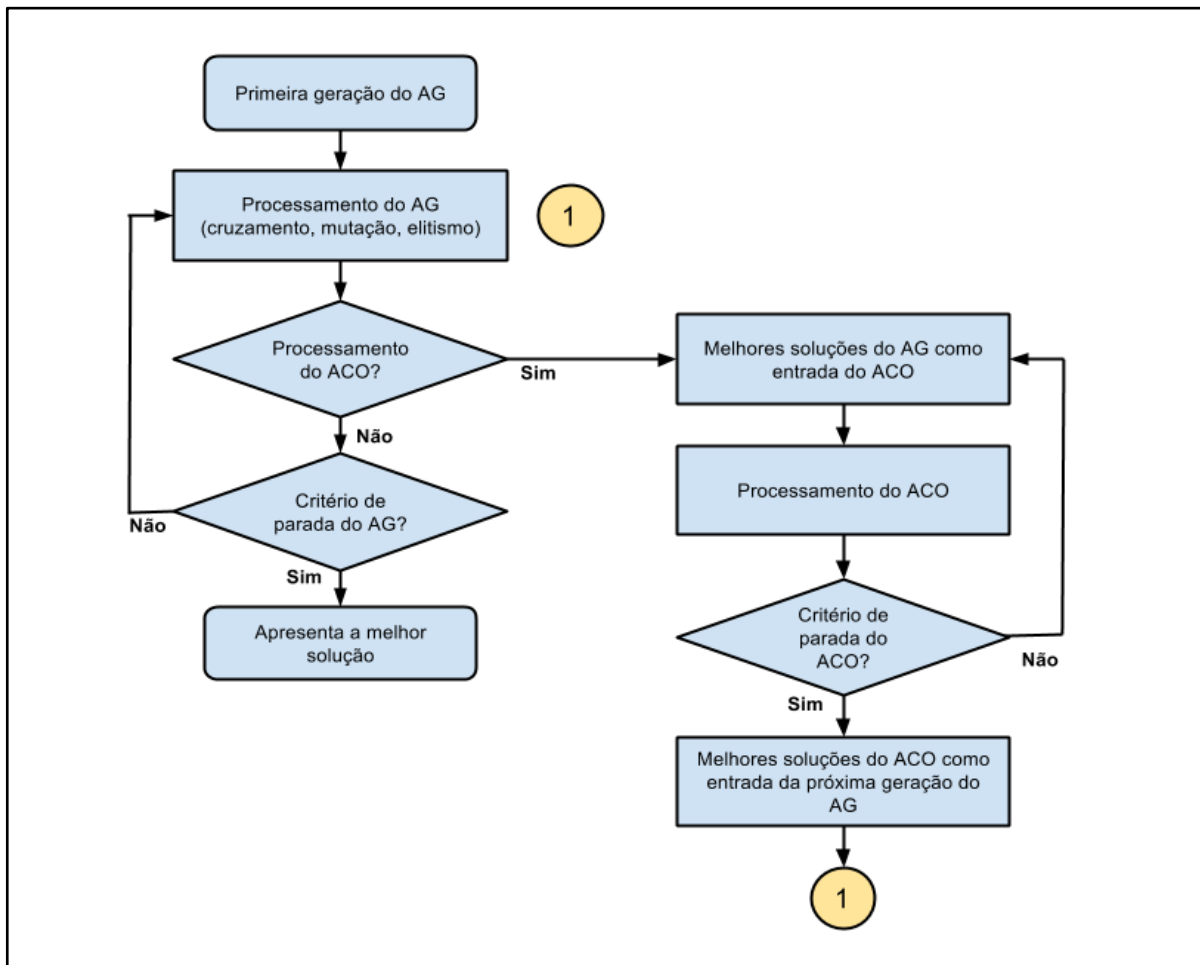
Foi utilizada a seguinte abordagem cooperativa de combinação, baseada na abordagem utilizada por Carvalho (2007):

- O processamento do AG foi definido para  $n$  gerações;
- A cada  $m$  gerações os melhores indivíduos do AG são utilizados como entrada para o ACO;
- O ACO é executado por  $i$  iterações;
- O melhor resultado do ACO é utilizado como nova entrada para o AG que continua a ser executado até a geração  $n$ .

O fluxograma da Figura 14 mostra, de forma simplificada, as fases do processo de combinação cooperativa entre as meta-heurísticas.



**Figura 14** – Fluxograma das fases do processo de combinação cooperativa entre as meta-heurísticas



Fonte: o autor.

Nesta fase ainda foram comparados os resultados da aplicação das heurísticas construtivas individualmente, em relação a aplicação das mesmas como auxiliares das meta-heurísticas, de forma a identificar se os resultados justificam a utilização das meta-heurísticas.

#### - FASE 5: AJUSTES NAS IMPLEMENTAÇÕES E ALTERAÇÕES NA COMBINAÇÃO DAS TÉCNICAS

A partir dos resultados obtidos nas fases anteriores, foram realizados ajustes nas implementações das meta-heurísticas individuais e na abordagem de combinação entre elas. Foi incluída uma heurística de busca local no ACO. Esta busca local funciona da seguinte forma:

- a) Após a atualização de feromônios, é selecionado um percentual dos itens das soluções para serem rotacionados;

- b) Este percentual é o mesmo utilizado no operador de mutação do AG, com objetivo a equiparar o ACO com o AG, que utiliza o operador de mutação na tentativa de melhorar as soluções existentes.
- c) Após aplicação da rotação nos itens selecionados, as soluções são novamente avaliadas e, caso apresentem melhores resultados do que o anterior a aplicação da rotação, atualiza-se novamente a matriz de feromônios.

Nesta fase, foram utilizados os mesmos 4 problemas tratados nas Fases 2, 3 e 4 (P1, P2, P3 e P4) e novamente aplicadas as meta-heurísticas separadas (AG e ACO) e combinadas (AG+ACO), de forma a definir a melhor delas, para aplicação nas fases posteriores (Fases 6 e 7).

#### - FASE 6: APLICAÇÃO DAS META-HEURISTICAS EM INSTÂNCIAS DA LITERATURA

Nesta fase, a meta-heurística vencedora (AG, ACO ou AG+ACO) da Fase 5 foi aplicada nas instâncias das Classes 1 a 10 tratadas por Lodi et al. (1999). Estas instâncias são divididas em 10 classes de tamanhos variados de itens e objetos.

As classes 1 a 6 foram apresentadas por Berkey e Wang (1987) e as classes 7 a 10 por Lodi et al. (1999). Apesar de se utilizar as instâncias de Berkey e Wang, os resultados considerados foram os de Lodi et al., aplicados a estas instâncias, por tratarem o corte bidimensional guilhotinado com rotação dos itens.

Cada uma das 10 classes é dividida em 5 subclasses com 10 instâncias de problemas cada, variando-se a quantidade de itens (20, 40, 60, 80 e 100), totalizando 500 instâncias. A avaliação de cada instância (AI) foi feita a partir dos resultados de aplicação da Equação 11.

$$AI = \frac{LB}{N} \quad (11)$$

Na qual:

*LB* é o *Lower Bound* do problema tratado.

*N* é a quantidade de chapas ou objetos utilizados na solução.

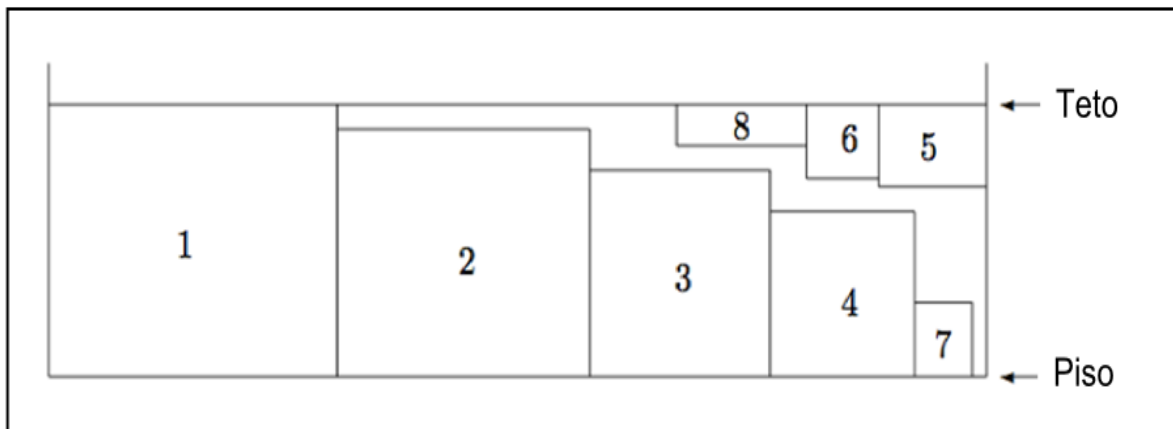
Para definição do *Lower Bound* de cada problema é calculado o somatório das áreas de todos os itens do problema, este valor é, então, dividido pela área do objeto utilizado, para encontrar o número mínimo teórico (arredondado para cima) de objetos necessários para acomodar todos os itens.

O objetivo de utilizar tais instâncias da literatura é o de identificar de que forma a combinação do AG com ACO pode melhorar os resultados de técnicas aplicadas anteriormente, e que foram denominadas de FCRG e KPRG, por Lodi et al. (1999).

As heurísticas utilizadas por Lodi et al. são a *Floor-Ceiling* (FC), que é uma abordagem que estende a maneira como os itens são acondicionados em níveis e a *Knapsack Problem* (KP) que é baseada no problema da mochila. Foram denominadas de FCRG e KPRG quando aplicadas a problemas de corte bidimensional guilhotinado (G) e com rotação dos itens (R).

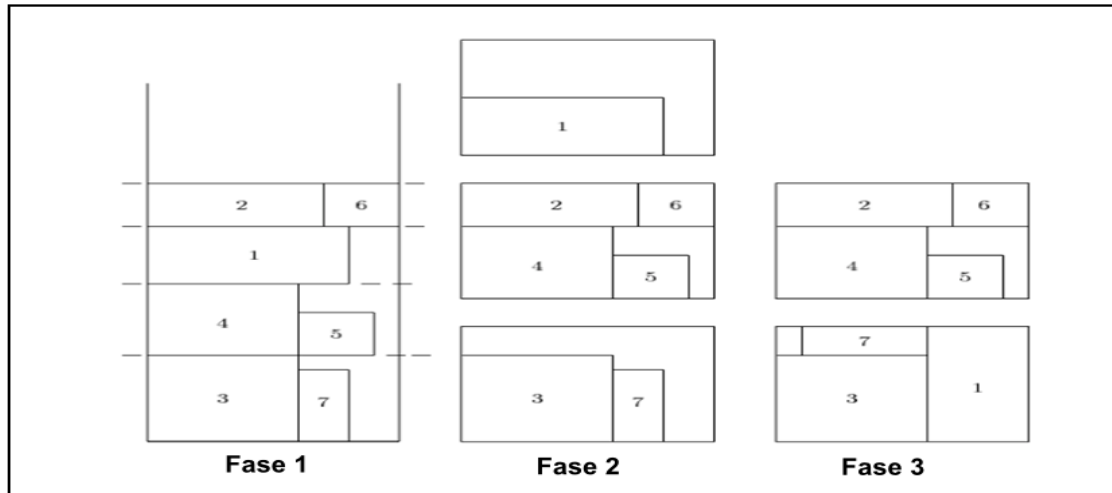
A heurística FC adiciona os itens, da esquerda para a direita, com a sua extremidade inferior no piso do nível corrente. Em seguida, adiciona os itens restantes, da direita para a esquerda, com a sua borda superior no teto do nível corrente. Os itens adicionados no teto só podem ser aqueles que não puderam ser adicionados na parte inferior, conforme pode ser visto na Figura 15.

**Figura 15** – Aplicação da heurística construtiva FC.



Fonte: Adaptado de Lodi et al. (1999).

A heurística KP, primeiro encaixa os itens como no problema da mochila, em seguida separa os níveis conforme o tamanho dos objetos e aplica rotações nos itens de forma a otimizar a solução inicial, conforme mostrado na Figura 16.

**Figura 16** – Aplicação da heurística KP.

Fonte: Lodi et al. (1999).

#### - FASE 7: BENCHMARKING COM SOFTWARES COMERCIAIS DE OTIMIZAÇÃO DO CORTE BIDIMENSIONAL

As meta-heurísticas individuais (AG e ACO) ou combinadas (AG+ACO) que apresentaram os melhores resultados na Fase 5 foram aplicadas em mais 16 problemas selecionados da base de dados de 50 pedidos, e os resultados foram comparados com os resultados apresentados por dois *softwares* comerciais, o Opty Way e o Lisec GPS.opt aplicados nos mesmos problemas (8 problemas para o Opty Way e 8 problemas para o Lisec GPS.opt).

Os problemas escolhidos foram denominados de P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17, P18, P19 e P20 (Tabela 2 e Tabela 3). Para o processamento dos problemas P5 à P20 foram utilizados objetos de 6,0 x 3,21 metros, que se referem ao tamanho de chapa padrão utilizado nos pedidos de corte que constam na base de dados.

**Tabela 2** – Informações dos problemas P5 à P12.

Problema	Tamanho da chapa matriz	Quantidade de peças para cortar
P5	3210 x 6000	187
P6	3210 x 6000	379
P7	3210 x 6000	58
P8	3210 x 6000	255
P9	3210 x 6000	255
P10	3210 x 6000	324
P11	3210 x 6000	103
P12	3210 x 6000	167

**Tabela 3** – Informações dos problemas P13 à P20.

Problema	Tamanho da chapa matriz	Quantidade de peças para cortar
P13	3210 x 6000	50
P14	3210 x 6000	78
P15	3210 x 6000	13
P16	3210 x 6000	65
P17	3210 x 6000	33
P18	3210 x 6000	102
P19	3210 x 6000	94
P20	3210 x 6000	69

No próximo capítulo (4) serão apresentados e discutidos os resultados obtidos em cada uma das 7 fases.

## 4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Neste capítulo apresenta-se a realização dos experimentos e discutem-se os resultados. Como descrito no capítulo 3 a realização dos experimentos foi dividida em sete fases:

- **Fase 1:** Preparatória;
- **Fase 2:** Implementação e aplicação do AG;
- **Fase 3:** Implementação e aplicação do ACO;
- **Fase 4:** Implementação e aplicação das meta-heurísticas combinadas;
- **Fase 5:** Ajustes nas implementações e alterações na combinação das técnicas;
- **Fase 6:** Aplicação das meta-heurísticas em instâncias da literatura;
- **Fase 7:** Benchmarking com softwares comerciais.

Para o processamento dos problemas P1a P20 foram utilizados objetos de 6,0 x 3,21 metros, que se referem ao tamanho de chapa padrão utilizado nos pedidos de corte que constam na base de dados utilizada.

Os atributos contidos nas Tabelas 4, 5 e 6 que mostram os resultados da aplicação do AG e do ACO individualmente e combinadas são:

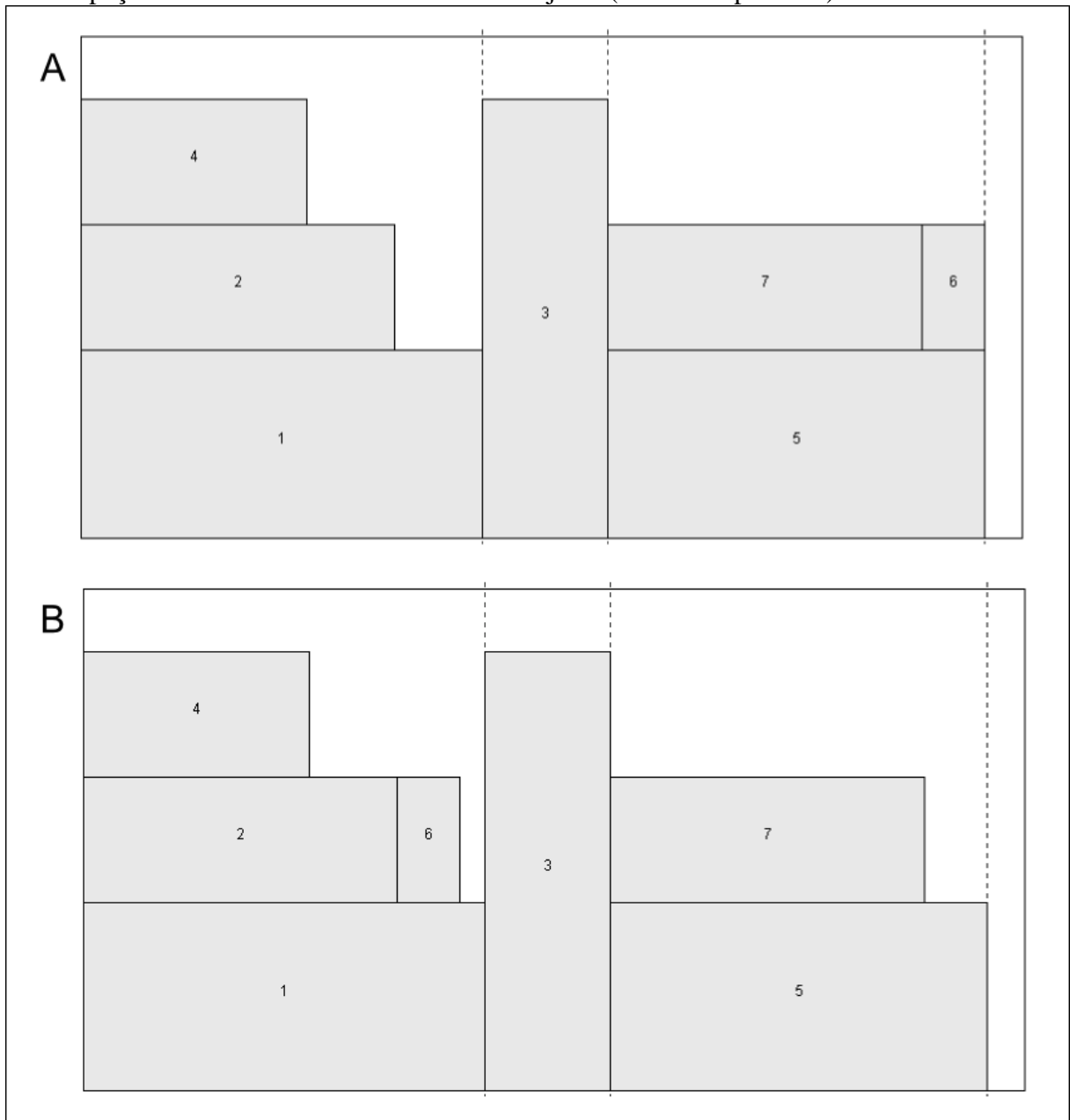
- Problema: P1, P2, P3 e P4;
- Itens: quantidade de itens de P1, P2, P3 e P4;
- Objetos necessários: quantidade de chapas necessárias para cortar os itens;
- tempo médio de processamento em minutos: é o tempo necessário para execução dos algoritmos aplicados na resolução de cada problema.

### 4.1 REALIZAÇÃO DOS EXPERIMENTOS

#### - FASE 1: PREPARATÓRIA

Nesta fase, após a seleção dos problemas da base de dados foram implementadas as heurísticas FFDWDH e BFDWDH. Foram validadas após a implementação utilizando um experimento baseado na demonstração de Lodi et al. (1999), conforme pode ser visto na Figura 17.

**Figura 17** – Validação da implementação das heurísticas BFDWDH e FFDWDH, onde na FFDWDH (B), a peça 6 foi encaixada na primeira área disponível e na BFDWDH (A) a mesma peça foi encaixada na área com melhor ajuste (menor desperdício).



Fonte: o autor.

Devido à heurística BFDWDH privilegiar o melhor aproveitamento das áreas livres em relação à heurística FFDWDH, apenas a BFDWDH foi escolhida para se utilizada nos experimentos das próximas fases.

## - FASE 2: IMPLEMENTAÇÃO E APLICAÇÃO DO AG

Por meio dos testes de calibração concluiu-se que o AG com os seguintes parâmetros apresentou melhores resultados:

- Número de gerações: 200
- Número de cromossomos: 100
- Percentual de mutação: 1%
- Percentual de Elitismo: 5%
- Pontos de corte: 2

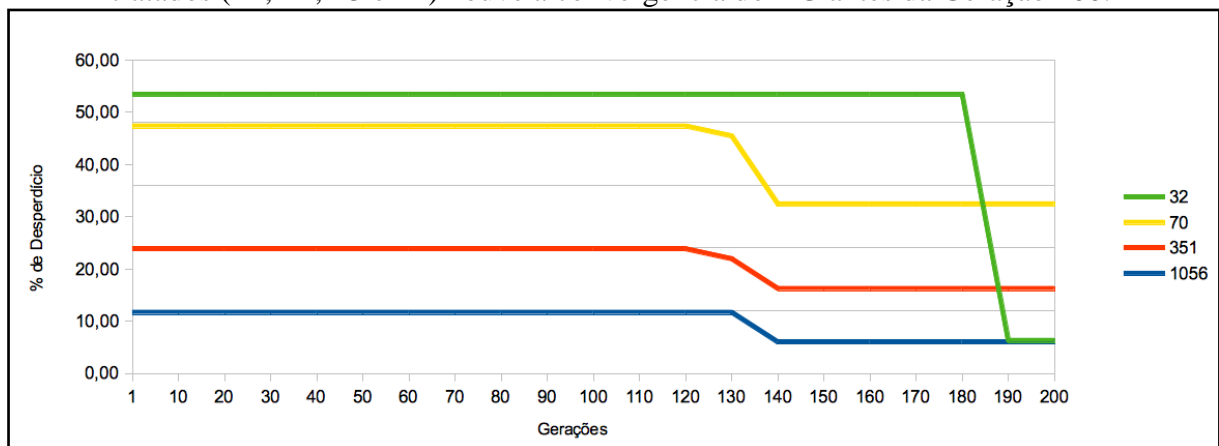
Estes parâmetros foram utilizados na aplicação com o AG. Os resultados da aplicação do AG nos problemas P1, P2, P3 e P4 podem ser vistos na Tabela 4.

**Tabela 4-** Resultados da aplicação do AG nos problemas P1, P2, P3 e P4.

Problema	Itens	Objetos necessários	Tempo médio de processamento (minutos)
P1	32	1	0,05 min.
P2	70	5	0,1 min.
P3	351	24	1,2 min.
P4	1056	68	10,2 min.

Na Figura 18 pode-se ver o gráfico que justifica a escolha de 200 gerações para o AG, onde, no processamento dos 4 problemas tratados (P1 a P4), em todos eles houve a convergência do AG antes da geração 200.

**Figura 18** – Calibração do AG em função do número de gerações, onde nos 4 problemas tratados (P1, P2, P3 e P4) houve a convergência do AG antes da Geração 200.



Fonte: o autor.



### - FASE 3: IMPLEMENTAÇÃO E APLICAÇÃO DO ACO

Por meio dos testes de calibração concluiu-se que o ACO com os seguintes parâmetros apresentou melhores resultados:

- Número de formigas: 15.
- Número de iterações: 10.
- Valor de Alfa : 2.
- Valor de Beta : 5.

Estes parâmetros foram utilizados na aplicação com o ACO. Os resultados da aplicação do ACO nos problemas P1, P2, P3 e P4 podem ser vistos na Tabela 5.

**Tabela 5-** Resultados da aplicação do ACO nos problemas P1, P2, P3 e P4.

<b>Problema</b>	<b>Itens</b>	<b>Objetos necessários</b>	<b>Tempo médio de processamento (minutos)</b>
P1	32	2	0,03 min.
P2	70	6	0,9 min.
P3	351	24	1,4 min.
P4	1056	68	19,8 min.

### - FASE 4 – IMPLEMENTAÇÃO E APLICAÇÃO DAS META-HEURÍSTICAS COMBINADAS

Na aplicação das técnicas combinadas, foram utilizados os seguintes parâmetros para o ACO:

- Número de formigas: 15.
- Número de iterações: 10.
- Valor de Alfa : 2.
- Valor de Beta : 5.

Na aplicação das técnicas combinadas, foram utilizados os seguintes parâmetros para o AG:

- Número de gerações: 100.
- Número de cromossomos: 100.
- Percentual de mutação: 1%.
- Percentual de Elitismo: 5%.
- Pontos de corte: 2.

Na aplicação das técnicas combinadas, foi reduzido o número de gerações do AG com o objetivo de manter o tempo de processamento do algoritmo híbrido próximo dos tempos de processamento das meta-heurísticas aplicadas individualmente.

Os resultados da aplicação das meta-heurísticas combinadas nos problemas P1, P2, P3 e P4 podem ser vistos na Tabela 6.

**Tabela 6-** Resultados da aplicação das meta-heurísticas combinadas nos problemas P1, P2, P3 e P4.

<b>Problema</b>	<b>Itens</b>	<b>Objetos necessários</b>	<b>Tempo médio de processamento (minutos)</b>
P1	32	1	0,05 min.
P2	70	5	0,9 min.
P3	351	24	1,7 min.
P4	1056	67	25,9 min.

### **- DISCUSSÃO DOS RESULTADOS**

Na comparação dos resultados das Fases 2, 3 e 4 foram considerados os seguintes critérios:

- Quantidade de objetos necessários para resolução dos problemas.
- Tempo de processamento decorrido para resolução dos problemas.

A comparação dos resultados de cada fase em relação à quantidade de objetos necessários para cada problema pode ser visualizada na Tabela 7.

**Tabela 7** - Comparação dos resultados de cada fase em relação à quantidade de objetos necessários para cada problema.

<b>FASE</b>	<b>Quantidade de objetos utilizados para cada problema</b>			
	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>
FASE 2 (AG)	1	5	24	68
FASE 3 (ACO)	2	6	24	68
FASE 4 (AG+ACO)	1	5	24	67

Conforme pode ser visto na Tabela 7, em termos de qualidade das soluções, ou seja, a utilização do menor número possível de objetos para resolução do problema, as três fases apresentaram resultados que beiram igualdade ou semelhança em todos os problemas. Vale

destacar que as meta-heurísticas combinadas da Fase 4 foram sensivelmente melhores do que o ACO nos P1, P2 e P4 com empate no P3. No caso do AG as meta-heurísticas combinadas foram melhores apenas no P4.

A comparação dos resultados de cada fase em relação ao tempo de processamento de cada problema pode ser visualizada na Tabela 8.

**Tabela 8-** Comparação dos resultados de cada fase em relação ao tempo de processamento de cada problema.

FASE	Tempo de processamento médio de cada problema (minutos)			
	P1	P2	P3	P4
FASE 2 (AG)	0,05	0,1	1,2	10,2
FASE 3 (ACO)	0,03	0,9	1,4	19,8
FASE 4 (AG+ACO)	0,05	0,9	1,7	25,9

Conforme pode ser visto na Tabela 8, a comparação dos resultados de cada fase, em relação ao tempo de processamento dos problemas, mostrou que o AG teve melhor desempenho do que o ACO e do que as meta-heurísticas combinadas nos P2 e P3 e, no P4 atingiu a metade do tempo de processamento do ACO e menos da metade no caso das meta-heurísticas combinadas. O ACO se saiu melhor somente no P1 e as meta-heurísticas combinadas empataram com o AG no P1 se saindo pior nos problemas P2, P3 e P4.

Considerando os resultados apresentados na Tabela 7 e os resultados apresentados na Tabela 8 se verifica que o ACO teve o pior desempenho seguido das meta-heurísticas combinadas, apesar do bom resultado apresentado no problema P4 (Tabela 8) e o AG apresentou até essa fase melhores resultados em função dos menores tempos de processamento.

Ainda nesta fase, foram comparados os resultados das heurísticas construtivas (*First-fit* e *Best-fit*) sozinhas em relação a utilização das mesmas como auxiliares das Meta-heurísticas (AG+ACO). Os resultados são melhores quando se utilizam as heurísticas como auxiliares das meta-heurísticas, conforme Tabela 9.

**Tabela 9** - Comparação dos resultados das heurísticas individuais em relação as mesmas como auxiliares das meta-heurísticas combinadas.

	Número de peças	<i>First-fit e Best-fit</i>	AG+ACO
<b>Percentual de perda nas chapas utilizadas</b>	1056	32,28%	27,83%
	351	16,72%	9,48%
	32	53,15%	6,30%

**- FASE 5: AJUSTES NAS IMPLEMENTAÇÕES E ALTERAÇÕES NA COMBINAÇÃO DAS TÉCNICAS**

Conforme pode ser visto na Tabela 10, em termos de qualidade das soluções, o AG+ACO apresentou melhor resultado em relação as duas meta-heurísticas aplicadas individualmente apenas para P4. O ACO, após a inclusão da busca local, apresentou resultados semelhantes ao AG para os 4 problemas tratados.

**Tabela 10** - Comparação dos resultados das meta-heurísticas individuais em relação as mesmas combinadas, após os ajustes da Fase 5.

META-HEURÍSTICA	Quantidade de objetos utilizados para cada problema					Média
	P1	P2	P3	P4		
AG	1	5	24	68	24,5	
ACO	1	5	24	68	24,5	
AG+ACO	1	5	24	67	24,2	

Em relação ao tempo de processamento, o AG se mostrou melhor, conforme pode ser visto na Tabela 11, ficando o ACO com o segundo melhor tempo e o AG+ACO apresentou e maior tempo de processamento.

**Tabela 11** - Comparação do tempo de processamento das meta-heurísticas individuais em relação as mesmas combinadas.

META-HEURÍSTICA	Tempo de processamento médio de cada problema (minutos)					Média
	P1	P2	P3	P4		
AG	0,05	0,2	1,3	12,3	3,46	
ACO	0,05	1,1	1,9	16,6	4,91	
AG+ACO	0,06	1,3	1,9	18,8	5,51	

Apesar do AG ter apresentado os melhores tempos de processamento (Tabela 11), como a diferença dos tempos de processamento entre o AG, o ACO e o AG+ACO é pequena (cerca de 2 minutos na média) em relação aos requisitos da aplicação real das soluções, foi considerado o AG+ACO melhor por ter apresentado, na média geral, um melhor aproveitamento (1,22 % maior) dos objetos utilizados.

Considerando-se um consumo médio diário de 200 chapas, os resultados obtidos com a aplicação do AG+ACO (uma média de aproveitamento 1,22% maior que o AG e o ACO), representariam um aproveitamento de cerca de 2,44 chapas, o que equivale a uma economia de aproximadamente 47 metros quadrados de vidro por dia.

#### - FASE 6: APLICAÇÃO DAS META-HEURISTICAS EM INSTÂNCIAS DA LITERATURA

O AG+ACO, por apresentar melhor aproveitamento comparado as meta-heurísticas individuais, foi escolhido para ser aplicado nas instâncias da literatura (LODI et al.,1999). Os Quadros 2 a 7 apresentam a comparação entre AG+ACO e duas heurísticas (FCRG e KPRG) apresentadas por Lodi et al (1999) nas instâncias de Berkey e Wang (1987). As análises foram realizadas considerando a média dos resultados dos itens de cada uma das 10 classes.

**Quadro 3** – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 1

Classe	Itens	AG+ACO	FCRG	KPRG
1	20	1,06	1,06	1,06
	40	1,07	1,08	1,07
	60	1,09	1,09	1,07
	80	1,08	1,09	1,08
	100	1,07	1,07	1,05
	<b>Media</b>		1,075	1,078

Verifica-se no Quadro 3 que o AG+ACO apresentou melhor média do que a heurística FCRG, porém não foi melhor que a heurística KPRG.

**Quadro 4** – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 2

Classe	Itens	AG+ACO	FCRG	KPRG
2	20	1,00	1,00	1,00
	40	1,10	1,10	1,10
	60	1,10	1,05	1,15
	80	1,07	1,03	1,07
	100	1,03	1,03	1,03
	<b>Média</b>		1,060	1,042

Verifica-se no Quadro 4 que o AG+ACO apresentou melhor média que a heurística KPRG e pior que a heurística FCRG para as instâncias da Classe 2.

**Quadro 5** – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 3

Classe	Itens	AG+ACO	FCRG	KPRG
3	20	1,10	1,18	1,12
	40	1,15	1,16	1,16
	60	1,12	1,19	1,12
	80	1,12	1,15	1,12
	100	1,09	1,13	1,10
	<b>Média</b>		1,116	1,162

Verifica-se no Quadro 5 que o AG+ACO apresentou melhor média para as instâncias da Classe 3 que as heurísticas FCRG e KPRG.

**Quadro 6** – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 4

Classe	Itens	AG+ACO	FCRG	KPRG
4	20	1,00	1,00	1,00
	40	1,00	1,00	1,00
	60	1,10	1,10	1,10
	80	1,10	1,10	1,10
	100	1,06	1,07	1,07
	<b>Média</b>		1,052	1,054

Verifica-se no Quadro 6 que o AG+ACO apresentou melhor média para as instâncias da Classe 4 que as heurísticas FCRG e KPRG.

**Quadro 7** – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 5

Classe	Itens	AG+ACO	FCRG	KPRG
5	20	1,08	1,08	1,08
	40	1,11	1,10	1,11
	60	1,10	1,11	1,11
	80	1,11	1,11	1,10
	100	1,09	1,10	1,09
	<b>Media</b>		1,099	1,100

Verifica-se no Quadro 7 que o AG+ACO quando aplicado às instâncias da Classe 5 apresentou melhor média que a heurística FCRG, porém não foi melhor que a heurística KPRG.

**Quadro 8** – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 6

Classe	Itens	AG+ACO	FCRG	KPRG
6	20	1,00	1,00	1,00
	40	1,40	1,40	1,40
	60	1,05	1,05	1,05
	80	1,00	1,00	1,00
	100	1,07	1,07	1,10
	<b>Média</b>		1,103	1,104

Verifica-se no Quadro 8 que o AG+ACO quando aplicado às instâncias da Classe 6 apresentou melhor média que as heurísticas FCRG e KPRG.

Os Quadros 9 a 12 apresentam a comparação entre AG+ACO e duas heurísticas apresentadas por Lodi et al (1999) nas instâncias das Classes 7 a 10.

**Quadro 9** – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 7

Classe	Itens	AG+ACO	FCRG	KPRG
7	20	1,13	1,19	1,17
	40	1,18	1,17	1,17
	60	1,15	1,18	1,16
	80	1,17	1,17	1,17
	100	1,17	1,17	1,16
	<b>Média</b>		1,161	1,176

Verifica-se no Quadro 9 que o AG+ACO quando aplicado às instâncias da Classe 7 apresentou melhor média que as heurísticas FCRG e KPRG.

**Quadro 10** – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 8

Classe	Itens	AG+ACO	FCRG	KPRG
8	20	1,12	1,16	1,16
	40	1,19	1,19	1,19
	60	1,18	1,18	1,18
	80	1,16	1,16	1,15
	100	1,16	1,17	1,17
	<b>Média</b>	1,161	1,172	1,170

Verifica-se no Quadro 10 que o AG+ACO quando aplicado às instâncias da Classe 8 apresentou melhor média que as heurísticas FCRG e KPRG.

**Quadro 11** – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 9

Classe	Itens	AG+ACO	FCRG	KPRG
9	20	1,00	1,00	1,00
	40	1,00	1,01	1,01
	60	1,00	1,01	1,01
	80	1,01	1,01	1,01
	100	1,00	1,01	1,01
	<b>Média</b>	1,002	1,008	1,008

Verifica-se no Quadro 11 que o AG+ACO quando aplicado às instâncias da Classe 9 apresentou melhores resultados que as heurísticas FCRG e KPRG.

**Quadro 12** – Comparativo entre AG+ACO e duas heurísticas em instâncias da Classe 10

Classe	Itens	AG+ACO	FCRG	KPRG
10	20	1,10	1,15	1,12
	40	1,06	1,09	1,09
	60	1,10	1,09	1,08
	80	1,06	1,06	1,06
	100	1,07	1,07	1,05
	<b>Média</b>	1,077	1,092	1,080



Verifica-se no Quadro 12 que o AG+ACO quando aplicado às instâncias da Classe 10 apresentou melhores resultados que as heurísticas FCRG e KPRG.

No Quadro 13 é apresentada a comparação das médias de cada classe e da média geral do AG+ACO com as heurísticas FCRG e KPRG.

**Quadro 13** – Comparação das médias de cada classe e da média geral do AG+ACO com as heurísticas FCRG e KPRG

Classe	AG+ACO	FCRG	KPRG
1	1,075	1,078	1,066 (melhor)
2	1,060	1,042 (melhor)	1,070
3	1,116 (melhor)	1,162	1,124
4	1,052 (melhor)	1,054	1,054
5	1,099	1,100	1,098 (melhor)
6	1,103 (melhor)	1,104	1,110
7	1,161 (melhor)	1,176	1,166
8	1,161 (melhor)	1,172	1,170
9	1,002 (melhor)	1,008	1,008
10	1,077 (melhor)	1,092	1,080
<b>Média Geral</b>	1,091 (melhor)	1,099	1,095

No Quadro 13 pode-se visualizar que a média geral (1,091) foi sensivelmente superior as médias das heurísticas FCRG (1,099) e KPRG (1,095). Pode-se verificar ainda que das dez classes tratadas, o AG+ACO teve melhores resultados em sete classes, o que comprova que a aplicação das meta-heurísticas bioinspiradas combinadas (AG+ACO) podem apresentar bons resultados.

#### - FASE 7: BENCHMARKING COM SOFTWARES COMERCIAIS DE OTIMIZAÇÃO DO CORTE BIDIMENSIONAL

Os resultados do AG+ACO (Fase 5) foram comparados com os resultados obtidos dos seguintes softwares comerciais de otimização de corte Opty Way e Lisec GPS.opt aplicados na resolução de 8 problemas cada, totalizando 16 problemas.

No Quadro 14 se pode verificar a comparação dos resultados em quantidade de chapas utilizadas da aplicação do *software* Lisec GPS.opt e do AG+ACO nos problemas P5 à P12.

**Quadro 14** – Comparação dos resultados (quantidade de chapas utilizadas) da aplicação do Lisec GPS.opt e do AG+ACO nos problemas P5 a P12

<b>Problema</b>	<b>Lisec GPS.opt</b>	<b>AG+ACO</b>
P5	12	13
P6	22	23
P7	5	5
P8	11	11
P9	11	11
P10	11	11
P11	10	11
P12	13	14
<b>Média</b>	11,88	12,38

Verifica-se no Quadro 14 que os resultados do AG+ACO quando aplicados nos problemas P5a P12 foram piores do que os resultados apresentados pelo software Lisec GPS.opt.

No Quadro 15 se pode verificar a comparação dos resultados em quantidade de chapas utilizadas da aplicação do *software* Opty Way e do AG+ACO nos problemas P13 a P20.

**Quadro 15** – Comparação dos resultados (quantidade de chapas utilizadas) da aplicação do Opty Way e do AG+ACO nos problemas P13 a P20

<b>Problema</b>	<b>Opty Way</b>	<b>AG+ACO</b>
P13	4	4
P14	7	7
P15	4	5
P16	7	8
P17	3	4
P18	10	11
P19	25	25
P20	5	6
<b>Média</b>	8,13	8,75

Verifica-se no Quadro 15 que os resultados do AG+ACO quando aplicados nos problemas P13 a P20 foram piores do que os resultados apresentados pelo *software* Opty Way.

De uma forma geral, os resultados dos softwares comerciais aplicados nos problemas P5 a P20 foram melhores que os resultados do AG+ACO, que apresentou na média geral o consumo de aproximadamente meia chapa a mais.

Nesta fase não foi possível comparar os tempos de processamento do AG+ACO em relação ao Lisec GPS.opt e o Opty Way porque os resultados dos problemas da base de dados utilizada não possuem esta informação.

O próximo capítulo irá tratar das conclusões e das perspectivas futuras.

## 5 CONCLUSÕES E PERSPECTIVAS FUTURAS

O problema do corte bidimensional guilhotinado na indústria vidreira ocasiona uma das principais dificuldades observadas neste setor, que se refere à minimização do desperdício de materiais.

Neste trabalho, duas meta-heurísticas bioinspiradas, os Algoritmos Genéticos e os Algoritmos de Otimização Colônia de Formigas, foram aplicadas, individualmente e combinadas na solução do problema do corte bidimensional guilhotinado em uma indústria vidreira.

Para aplicar as meta-heurísticas escolhidas ao problema do corte bidimensional guilhotinado foram realizados experimentos divididos em sete fases:

- Fase 1: preparatória, foi selecionada a base de dados e implementadas as heurísticas e meta-heurísticas;
- Fase 2: o AG foi implementado e aplicado em 4 problemas da base de dados;
- Fase 3: ACO foi implementado e aplicado em 4 problemas da base de dados;
- Fase 4: o AG e o ACO foram combinados e aplicados em 4 problemas da base de dados;
- Fase 5: foram feitos ajustes na implementação do ACO e as meta-heurísticas foram novamente aplicadas, individualmente e combinadas, em 4 problemas da base de dados;
- Fase 6: o AG+ACO (que apresentou melhores resultados na Fase 5) foi aplicado em 500 instâncias da literatura e os resultados comparados com os resultados da aplicação das heurísticas FCRG e PKRG, apresentadas por Lodi et al. (1999);
- Fase 7: o AG+ACO foi aplicado em 16 problemas e os resultados foram comparados com os resultados apresentados por dois *softwares* comerciais, o Lisec GPS.opt e o Opty Way, aplicados aos mesmos problemas (8 problemas cada um).

Na comparação dos resultados das Fases 1, 2, 3 e 4 o ACO apresentou o pior desempenho seguido pelo AG+ACO, apesar do bom resultado deste apresentado no problema P4. O AG apresentou os melhores resultados considerando os menores tempos de processamento.

Na Fase 5, após a inclusão da heurística de busca local no ACO, o AG continuou apresentando sensíveis melhores tempos de processamento. Assim, a diferença dos tempos de processamento entre o AG, o ACO e o AG+ACO foi considerada pequena em relação aos requisitos do processo real de corte bidimensional na indústria vidreira. Desta forma, o

AG+ACO foi selecionado para ser aplicado nas Fases 6 e 7 pelo melhor aproveitamento geral em termos de minimização das chapas utilizadas.

Na Fase 6 o AG+ACO apresentou melhores resultados que as heurísticas FCRG e PKRG na média geral (1,091) e na análise das dez classes utilizadas com melhor desempenho em sete delas.

Na Fase 7 ao ter os seus resultados comparados com os softwares comerciais, o AG+ACO apresentou resultados inferiores aos apresentados pelo Opty Way e Lisec GPS.opt, consumindo na média geral cerca de meia chapa a mais do que estes. Vale ressaltar que o resultado obtido com o AG+ACO é considerado aceitável para os requisitos do problema tratado.

Concluiu-se, então, em resposta ao problema de pesquisa desse trabalho, que resultados positivos podem ser obtidos na solução do problema do corte bidimensional guilhotinado em uma indústria vidreira quando o Algoritmo Genético e o Algoritmo Colônia de Formigas são aplicados individualmente e, principalmente combinados.

Vale ressaltar que a aplicação do AG+ACO possui algumas limitações. Estas limitações foram evidenciadas na comparação com softwares comerciais, já que o AG+ACO não apresentou soluções tão boas quanto estes.

Pode-se atribuir parte destas limitações ao fato das empresas desenvolvedoras dos softwares comerciais de otimização do corte bidimensional não disponibilizarem informações que possibilitem a realização dos experimentos com maior profundidade, tais como o tipo de técnica utilizada nas otimizações, fato este que é compreensível, já que tais técnicas podem representar um diferencial em termos de competitividade industrial.

Além disso, esses softwares são ajustados para o tipo de problema da empresa, ou seja, são customizados. Esse fato reforça a importância dos resultados obtidos com o AG+ACO que não teve esse tipo de ajuste e, apesar disso, os resultados obtidos com as meta-heurísticas combinadas foram bem próximos dos resultados obtidos com os softwares comerciais, o que incentiva a continuação da investigação sobre o potencial das técnicas combinadas.

Considera-se a principal contribuição deste trabalho a metodologia experimental desenvolvida em sete fases envolvendo heurísticas, meta-heurísticas, instâncias da literatura e softwares comerciais e, que poderá servir de referência para estudos de continuidade por meio da exploração mais profunda e ou diversificada dos elementos citados. Sugere-se neste sentido associar a metodologia experimental a meta-heurística chamada Otimização por Enxame de Partículas ou PSO (do inglês *Particle Swarm Optimization*).

As pesquisas realizadas durante o desenvolvimento deste trabalho não têm a pretensão de esgotar o assunto, buscou-se realizar uma contribuição aos trabalhos existentes.

O trabalho tem continuidade ao considerar a implementação das heurísticas apresentadas por Lodi et al. e utilizá-las como auxiliares na combinação dos Algoritmos Genéticos, Algoritmo de Otimização Colônia de Formigas e outras meta-heurísticas como PSO e os Algoritmos de Abelhas (*Bees Algorithm*), de forma a demonstrar a contribuição dessas meta-heurísticas quando combinadas e utilizando-se as heurísticas construtivas como auxiliares.

## 5.1 CO-ORIENTAÇÃO EM PROJETO DE INICIAÇÃO CIENTÍFICA

Foi realizada co-orientação em projeto de Iniciação Científica, intitulado: “Combinação de Técnicas Meta-heurísticas Bio-inspiradas para Otimização do Corte Bidimensional Guilhotinado em uma Indústria Vidreira”.

Iniciado em 2012 e com conclusão em 2013, este projeto contou com a participação dos alunos de graduação Aleister Ferreira (bolsa PIBITI), André da Silva Souza, Lucas dos Santos Severino e Shirley Araujo Guimarães.

Os trabalhos publicados oriundos deste projeto foram:

1. COSTA, F. M.; SASSI, R. J. ; FERREIRA, A.; GUIMARÃES, S. A. Combinação de Técnicas meta-heurísticas bio-inspiradas para otimização do corte de chapas de vidro. In: Encontro Mineiro de Engenharia de Produção, 2012, Itajubá. Qualidade e Produtividade na Engenharia de Produção, 2012. v. VIII. p. 1-9.
2. COSTA, F. M. ; SASSI, R. J. ; Guimarães, S. A. Aplicação de Meta-heurísticas Bio-inspiradas Híbridas para Otimização do Corte Bidimensional Guilhotinado em uma Indústria Vidreira. In: XXXII Encontro Nacional de Engenharia de Produção (ENEGEP 2012), 2012, Bento Gonçalves. Anais do XXXII ENEGEP, 2012. v. 1. p. 1-10.

## 5.2 PUBLICAÇÕES DO AUTOR

### -Artigos publicados em periódicos

1. COSTA, F. M. ; Sassi, R. J. . Application of an hybrid bio-inspired meta-heuristic in the optimization of two-dimensional guillotine cutting in an Glass Industry. *Lecture Notes in Computer Science*, v. 7435, p. 802-809, 2012.
2. COSTA, F. M. ; Carvalho, T. V. ; Sassi, R. J. . Application of bio-inspired metaheuristics to guillotined cutting processes optimize in an glass industry. *Lecture Notes in Computer Science*, v. 8028, p. 407-413, 2013.

### -Artigos Publicados em Congressos

1. COSTA, F. ; SASSI, R. J. ; Guimarães, S.A. . Aplicação de Meta-heurísticas Bio-inspiradas Híbridas para Otimização do Corte Bidimensional Guilhotinado em uma Indústria Vidreira. In: XXXII Encontro Nacional de Engenharia de Produção (ENEGEP 2012), 2012, Bento Gonçalves. *Anais do XXXII ENEGEP*, v. 1. p. 1-10, 2012.
2. Ferreira, R. P. ; SASSI, R. J. ; COSTA, F. M. ; Ferreira, A. ; Souza, A. S. . Aplicando o Algoritmo de Otimização por Colônia de Formigas e os Mapas Auto-Organizáveis de Kohonen na Roteirização e Programação de Veículos. In: XVI Simpósio de Pesquisa Operacional e Logística da Marinha (SPOLM), 2012, Rio de Janeiro. *Defesa e Desenvolvimento Sustentável da Amazônia Azul*, v. 15. p. 1-12, 2012.

## REFERÊNCIAS

- AFFONSO, C. Aplicação de Redes *Neuro Fuzzy* ao Processamento de Polímeros na Indústria Automotiva. 2010. 110f. Dissertação (Mestrado em Engenharia de Produção) - Universidade Nove de Julho, São Paulo. 2010.
- ALVES, O.L.; GIMENEZ, I.F.; MAZALI, I.O. Vidros. **Cadernos Temáticos: Química Nova** Escola, Edição Especial, Fev. 2001.
- ASSIS, C.; TEIXEIRA, F. **O Vidro Plano no Brasil**. 1. ed. São Paulo: Câmara Brasileira do Livro, 2001.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR NM 294: Vidro float**. Rio de Janeiro, 2004. 11 p.
- BASTOS, E. A. Otimização de Seções Retangulares de Concreto Armado Submetidas à Flexo-compressão Oblíqua Utilizando Algoritmos Genéticos. 2004. 168f. Dissertação (Mestrado) - Universidade Federal do Rio de Janeiro, Ciências em Engenharia Civil, Rio de Janeiro, 2004.
- BELEW, R. K.; FORREST, S. *Learning and programming in classifier systems*. **Machine Learning**, v. 3, p. 193-223, Oct. 1988.
- BELOV, G.; Scheithauer, G. *A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting*. **European Journal of Operational Research**, v. 171, p. 85- 106, 2006.
- BERKEY, J. O.; WANG, P. Y. *Two dimensional finite bin packing algorithms*. **Journal of the Operation Research Society**, v. 38, p. 423-429, 1987.
- BERNARD, A. N. *Representation Selection for Constraint Satisfaction: A Case Study Using n-Queens*. **IEEE Expert**, v. 885, p. 16-23, 1990.
- CANTO, N. C. F.; SASSI, R. J.; COSTA, F.M. Aplicação de um algoritmo genético para solução do problema do corte bidimensional em lâminas de vidro. **CISTI'2010 (5 th Conferência Ibérica de Sistemas e Tecnologias de Informação)**, Santiago de Compostela, Espanha, v. 12, 2010.
- CAPRARA, A., TOTH, P. *Lower bounds and algorithms for the 2- dimensional vector packing problem*. **DiscreteAppl. Math**, v. 111, p. 231–262, 2001.
- CARVALHO, M. B. Aplicações de Meta-heurística Genética e *Fuzzy* no Sistema de Colônia de Formigas para o Problema do Caixeiro Viajante. 2007. Dissertação (Mestrado em Engenharia Elétrica ) – Universidade de Campinas (UNICAMP), Campinas , 2007.
- CARVALHO, M.F.; SILVA FILHO, O.S.; FERNANDES, C.A.O. O Planejamento da manufatura - Práticas Industriais e Métodos de Otimização. **Gestão e Produção**, v.5, n.1, p. 34-59, 1998.



- CASTRO, L. N.; ZUBEN, F.; J. VON. *Recent Developments in Biologically Inspired Computing*. Idea Group Publishing, 2005.
- CONSTANTINO, A. A.; NETTO, C. A. S.; ARAUJO, S. A. Problema de Escalonamento de Pessoal em Centrais de Atendimento Telefônico. **XXXVIII SBPO**, Goiania-GO, p. 1670-1681, Set. 2006.
- COSTA, F. M. ; CANTO, N. ; SASSI, R. J. *Study of the Application of Genetic Algorithms in Optimization of Cutting Glass Sheets*. **9<sup>th</sup> Industry Applications IEEE/IAS International Conference** (IEEEExplore Digital Library), São Paulo, v. 1, p. 1-3, 2010.
- COSTA, D.; HERTZ, A. *Ants can colour graphs*. *Journal of the Operational Research Society*, v. 48, p. 295-305, 1997.
- COSTA, F. M. ; SASSI, R. J. *Application of an hybrid bio-inspired meta-heuristic in the optimization of two-dimensional guillotine cutting in an Glass Industry*. *Lecture Notes in Computer Science*, v. 7435, p. 802-809, 2012.
- DORIGO, M.; GAMBARDELLA, L. M. *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*. *IEEE Transactions on Evolutionary Computation*, v. 1, n. 1, p. 53-66, 1997.
- DORIGO, M., STÜTZLE, T. *Ant Colony Optimization*. 1. ed. Massachusetts: The MIT Press, 2004. 305p.
- DROZDEK, A. *Estrutura de Dados e Algoritmos em C++*. reimpr. 1. ed. 2002, São Paulo: Cengage Learning, 2008. 579 p.
- DYCKHOFF, H. *A new linear programming approach to the cutting stock problem*. *Operations Research*, v. 29, p. 1092-1104, 1981.
- DYCKHOFF, H. *A typology of cutting and packing problems*. *European Journal of Operational Research*, v. 44(2), p. 145–159, 1990.
- EISENHARDT, K. M. *Building theories from case study research*. *Academy of management review*, v. 14.4, p. 532-550, 1989.
- FALKENAUER, E. *Genetic Algorithms and Grouping Problems*. 1.ed. New York: John Wiley & Sons Inc. 1998, 220p.
- GIL, A. C. **Como Elaborar Projetos de Pesquisa**. 4. ed. São Paulo: Atlas, 2002.
- GILMORE, P. C., GOMORY, R. E. *A linear programming approach to the cutting stock problem*. *Operations Research*, v. 9, p. 848-859, 1961.
- GILMORE, P. C., GOMORY, R. E. *A linear programming approach to the cutting stock problem - Part II*. *Operations Research*, v. 11, p. 863-888, 1963.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1.ed. MA: Addison-Wesley, 1989. 412 p.

GOLDSCHMIDT, R.; PASSOS, E. **Data Mining um guia prático: Conceitos, Técnicas, Ferramentas, Orientações e Aplicações**. 1. ed. , Rio de Janeiro: Ed. Campus, 2005.

GUANGDONG, H.; PING, L.; QUN, W. *A Hybrid Metaheuristic ACO-GA with an Application in Sports Competition Scheduling*. **8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing** (IEEEExplore Digital Library), 2007.

HOSEINI, P.; SHAYESTEH, M.G. *Hybrid Ant Colony Optimization, Genetic Algorithm, and Simulated Annealing for Image Contrast Enhancement*. **Evolutionary Computation (CEC)**, 2010.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. 1. ed. Ann Arbor: The University of Michigan Press, 1975.

HOPPER, E.; TURTON, B. *A Genetic Algorithm for a 2D Industrial Packing Problem*. **Computers & Industrial Engineering**, v. 37, p. 375-378, 1999.

KAUPA, P. H. Aplicação de Técnicas da Inteligência Artificial na Seleção de Ações para Investimento na Bolsa de Valores de São Paulo. 2013. 156f. Dissertação (Mestrado em Engenharia de Produção) - Universidade Nove de Julho, São Paulo, 2013.

KENNEDY, J.; EBERHART, R.C. *Particle swarm optimization*. **IEEE International conference on neural networks**, Perth, Austrália, p.1942-1948, 1995.

KOZA, J. R. **Genetic Programming: On the Programming of Computers by Means of Natural Selection**. Cambridge, MA: The MIT Press. 840 p, 1992.

LAWLER, E.L.; LENSTRA, J. K.; KAN, A. H. G. R.; SHMOYS, D. B. **The Traveling Salesman Problem**. 1.ed. Chichester: John Wiley & Sons, 1985.

LEVINE, J.; DUCATELLE, F. *Ant Colony Optimisation and Local Search for Bin Packing and Cutting Stock Problems*. **Journal of the Operational Research Society**, p. 93, 2003.

LODI, A.; MARTELLO, S.; VIGO, D. *Heuristic and Metaheuristic Approaches for a Class of Two-Dimensional Bin Packing Problems*. **Journal on Computing**, v.11, p.345-357, 1999.

LINDEN, R. **Algoritmos Genéticos: Aplicação em Bioinformática e no setor elétrico, programação Genética, Estratégias Evolucionárias e Algoritmos Meméticos**. 3.ed. Rio de Janeiro: Editora Ciência Moderna, 2012. 475 p.

LIU, B.; MENG, P. *Hybrid Algorithm Combining Ant Colony Algorithm with Genetic Algorithm for Continuous Domain*. Proceedings of the **9th International Conference for Young Computer Scientists (ICYCS)**, 2008.

MANTAWY, A. H.; ABDEL-MAGID, Y. L.; SELIM, S. Z. *Integrating Genetic Algorithms, Tabu Search, And Simulated Annealing For The Unit Commitment Problem*. **IEEE Transactionson Power Systems**. EUA: IEEE Press, v. 14, n. 3, p. 829-836, 1999.

MARTELLO, S., TOTH, P. *Knapsack Problems: Algorithms and Computer Implementations*, 1.ed. New York: John Wiley & Sons Inc, 1990.

MITCHELL, M. *An Introduction to Genetic Algorithms*, Bradford Book - MIT Press, 1996.

MIYAZAWA, F. K. Heurísticas e Metaheurísticas (Apostila). UNIVERSIDADE DE CAMPINAS - UNICAMP, INSTITUTO DE COMPUTAÇÃO - IC, Campinas, 2008. Disponível em: < <http://www.ic.unicamp.br/~fkm/lectures/heurísticas.pd> >. Acesso em: 14 nov. 2013.

MORABITO, R., ARENALES, M. N., ARCARO, V. F. *AND-OR-graph approach for two-dimensional cutting problems*. **European Journal of Operational Research**, v.58, p. 263-271, 1992.

MORABITO, R.; PUREZA, V. Geração de padrões de cortes bidimensionais guilhotinados restritos via programação dinâmica e busca em grafo-e/ou. **Produção**, v. 17, n. 1, p. 033-051, Jan./Abr. 2007.

PEDRO, M.F.; JOAQUIM JUNIOR, C.F.; TARRENTO, G.E. Alocação de mão-de-obra no planejamento de produção de bens de capital sob encomenda: UM ESTUDO DE CASO. **Tekhne e Logos**, Botucatu, SP, v.3, n.1., 2012. Disponível em: <<http://www.fatecbt.edu.br/seer/index.php/tl/article/view/6>>. Acesso em: 20 nov. 2013.

PEREIRA, A. J. V. Desenvolvimento de Novos Produtos em Vidro Utilizando Tecnologias de Prototipagem Rápida. 2006. Dissertação (Mestrado em Design Industrial) - Faculdade de Engenharia da Universidade do Porto, Escola Superior de Artes e Design de Matosinhos, Portugal, 2006.

PEREIRA, C. R.; COSTA, M. A. B. Um modelo de simulação de sistemas aplicado à programação da produção de um frigorífico de peixe. **Revista Produção Online**, Florianópolis, SC, v.12, n. 4, p. 972-1001, out./dez. 2012. Disponível em: <<http://www.producaoonline.org.br/rpo/article/view/994/962>>. Acesso em: 20 nov. 2013.

PERGHER, I.; SILVA, L. A., PACHECO, D. A. J.; VACCARO, G. L. R. Análise do impacto da variabilidade de fluxo no dimensionamento de KANBANS. **Revista Produção Online**, Florianópolis, SC, v.14, n. 1, p. 115-142, jan./mar. 2014. Disponível em: <<http://www.producaoonline.org.br/rpo/article/view/1542/1107>>. Acesso em 20 mar. 2014.

PINTO, R. F. Combinação de Técnicas da Inteligência Artificial para Previsão do Comportamento do Tráfego Veicular Urbano na Cidade de São Paulo. 2011. 108f. Dissertação (Mestrado em Engenharia de Produção) - Universidade Nove de Julho, São Paulo, 2011.

PRADO, D. F. M. Busca Tabu Aplicada ao Problema de Localização de Facilidades com Restrições de Capacidade e Fonte Única. 2007. 119f. Dissertação (Mestrado em engenharia Elétrica) - Universidade Estadual de Campinas, 2007.

PUREZA, V.; MORABITO, R. Uma heurística de busca tabu simples para o problema de carregamento de paletes do produtor. **Pesquisa Operacional**, Rio de Janeiro , v. 23, n. 2, Aug. 2003.

REIMANN, M.; STUMMER, M.; DOERNER, K. *A savings based Ant System for the vehicle routing problem*. Proceedings of the **Genetic and Evolutionary Computation Conference (GECCO-2002)**. Morgan Kaufman, p. 1317-1325, 2002.

ROCHA, M.I. Uma abordagem otimizada para o problema de alocação de equipes e escalonamento de tarefas para a obtenção de cronogramas eficientes. 2011. 120f. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual do Ceará, Fortaleza - CE, 2011.

RUSSEL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3. ed. Prentice-Hall, 2010. 1132p.

SASSI, R.J. Uma Arquitetura Híbrida para Descoberta de Conhecimento em Bases de Dados: Teoria dos *Rough Sets* e Redes Neurais Artificiais Mapas Auto-Organizáveis. 2006. 169 f. Tese (Doutorado em Engenharia Elétrica) – Universidade de São Paulo - USP, São Paulo, 2006.

SCHOPF, E. C.; SCHEPKE, C.; SILVA, M. L.; SILVA, P. F. Avaliação de Heurísticas de Melhoramento e da Metaheurística Busca Tabu para Solução de PRV. **VII Fórum de Tecnologias e XIV Simpósio Regional de Informática**, Santo Ângelo, 2004.

SEDGEWICK, R. **Algorithms in C++**. 3. ed., Pearson Education Inc. 2002.

SHERAFAT, H. Sistema Construtor de Circuitos e sua Aplicação na Roteirização de Coleta de Lixo Domiciliar. **Revista GEINTEC**. São Cristóvão/SE. v. 3, n. 5, p. 329-347, 2013.

SILVA, A. R. V. Uma nova modelagem para o problema de escalonamento de tarefas com restrições de recursos. 2006. 77 f. Dissertação (Mestrado em Computação) - Universidade Federal Fluminense. Niterói, 2006.

SILVA, J. L. C.; SOMA, N. Y. Um algoritmo polinomial para o problema de empacotamento de contêineres com estabilidade estática da carga. **Pesquisa Operacional**, Rio de Janeiro , v. 23, n. 1, Jan. 2003.

SOUZA, F.J. Modelos *Neuro-Fuzzy* Hierárquicos. Tese (Doutorado em Engenharia Elétrica) - Departamento de Engenharia Elétrica (PUC RIO), 1999.

TEMPONI, E. C. C. Uma Proposta de Resolução do Problema de Corte Bidimensional via Abordagem Metaheurística. 2007. Dissertação (Mestrado em Modelagem Matemática e Computacional) - CEFET/MG, Belo Horizonte, 2007.

STUTZLE, T.;HOOS, H. *MAX-MIN ant system*. **Future Generation Computer Systems**. p.889–914, 2000.

WASCHER, G., GAU, T. *Heuristics for the integer one-dimensional cutting stock problem: a computational study*. **OR Spektrum**, v. 18, p. 131-144, 1996.

WASCHER, G., HAUBNER, H., SCHUMANN, H. *An improved typology of cutting and packing problems*. **European Journal of Operational Research**, v. 183, p. 1109-1130, 2007.

XAVIER, E. C. ;MIYAZAWA, F. K. Algoritmos para Problemas de Empacotamento. Anais do **XXVII Congresso da SBC**, CTD, XX Concurso de Teses e Dissertações, Rio de Janeiro. p.1966 – 1973, 2007.

YIN, R. K. **Estudo de Caso**: planejamento e métodos. Trad. Daniel Grassi e Cláudio Damacena. 2.ed. Porto Alegre: Bookman, 2006. 205 p.

ZANAKIS, S.H.; EVANS, J. R. *Heuristic Optimization: Why, When, and How to Use It*. **Interfaces**, p.84-91, 1981.

ZHANG, D.; DU, L. *Hybrid Ant Colony Optimization Based on Genetic Algorithm for Container Loading Problem*. **International Conference of Soft Computing and Pattern Recognition (SoCPaR)**. p. 10-14, 2011.

## ANEXOS

### ANEXO A – Medidas dos itens da Classe VII.

Classe VII - 1		Classe VII - 2		Classe VII - 3		Classe VII - 4		Classe VII - 5	
Largura	Altura	Largura	Altura	Largura	Altura	Largura	Altura	Largura	Altura
75	56	9	90	32	85	20	73	48	97
27	98	6	99	26	25	31	84	18	71
24	74	4	77	32	93	37	77	46	9
34	99	97	11	46	96	55	78	49	94
1	80	50	74	36	86	31	84	22	82
43	84	13	84	97	16	33	71	33	67
6	88	1	72	42	98	4	93	25	96
6	89	8	97	21	94	13	76	19	94
38	98	53	74	1	94	53	54	7	15
21	78	81	66	83	19	98	2	8	71
49	75	10	76	14	76	33	82	66	90
93	31	33	94	8	100	82	65	1	68
37	90	19	90	29	76	39	97	4	78
73	21	61	68	19	88	21	70	46	84
4	71	30	89	47	96	23	79	20	99
35	93	19	100	70	20	74	26	44	84
48	89	45	72	78	2	35	47	13	87
12	90	4	88	23	72	8	82	45	69
79	13	13	74	42	36	75	76	34	78
42	10	77	19	17	66	55	99	49	74

Classe VII - 6		Classe VII - 7		Classe VII - 8		Classe VII - 9		Classe VII - 10	
Largura	Altura	Largura	Altura	Largura	Altura	Largura	Altura	Largura	Altura
7	70	48	67	43	68	27	78	39	25
95	12	35	11	39	72	22	99	6	76
42	90	16	96	42	67	86	49	48	37
48	99	10	98	20	84	45	91	72	9
86	43	12	92	19	67	90	7	10	84
31	92	39	11	5	81	45	84	42	6
21	85	35	90	23	69	44	95	82	44
42	87	20	68	43	97	24	80	2	70
15	41	37	35	6	73	5	70	42	92
31	84	4	82	33	98	80	68	40	72
11	32	16	92	17	80	20	99	74	12
48	83	41	80	27	84	15	78	6	76
11	66	24	96	55	83	37	78	37	74
43	27	32	99	81	34	28	80	50	82
13	98	9	67	22	99	7	77	13	42
16	87	25	66	22	19	18	26	21	90
19	81	22	75	17	76	30	95	43	49
16	92	21	89	24	88	37	89	80	1
27	82	74	29	89	86	6	97	25	89
45	99	19	43	29	23	33	97	48	78

Esta e as demais classes podem ser encontradas em:

<[http://www.or.deis.unibo.it/research\\_pages/ORinstances](http://www.or.deis.unibo.it/research_pages/ORinstances)>. Acesso em: 20 mar. 2014.

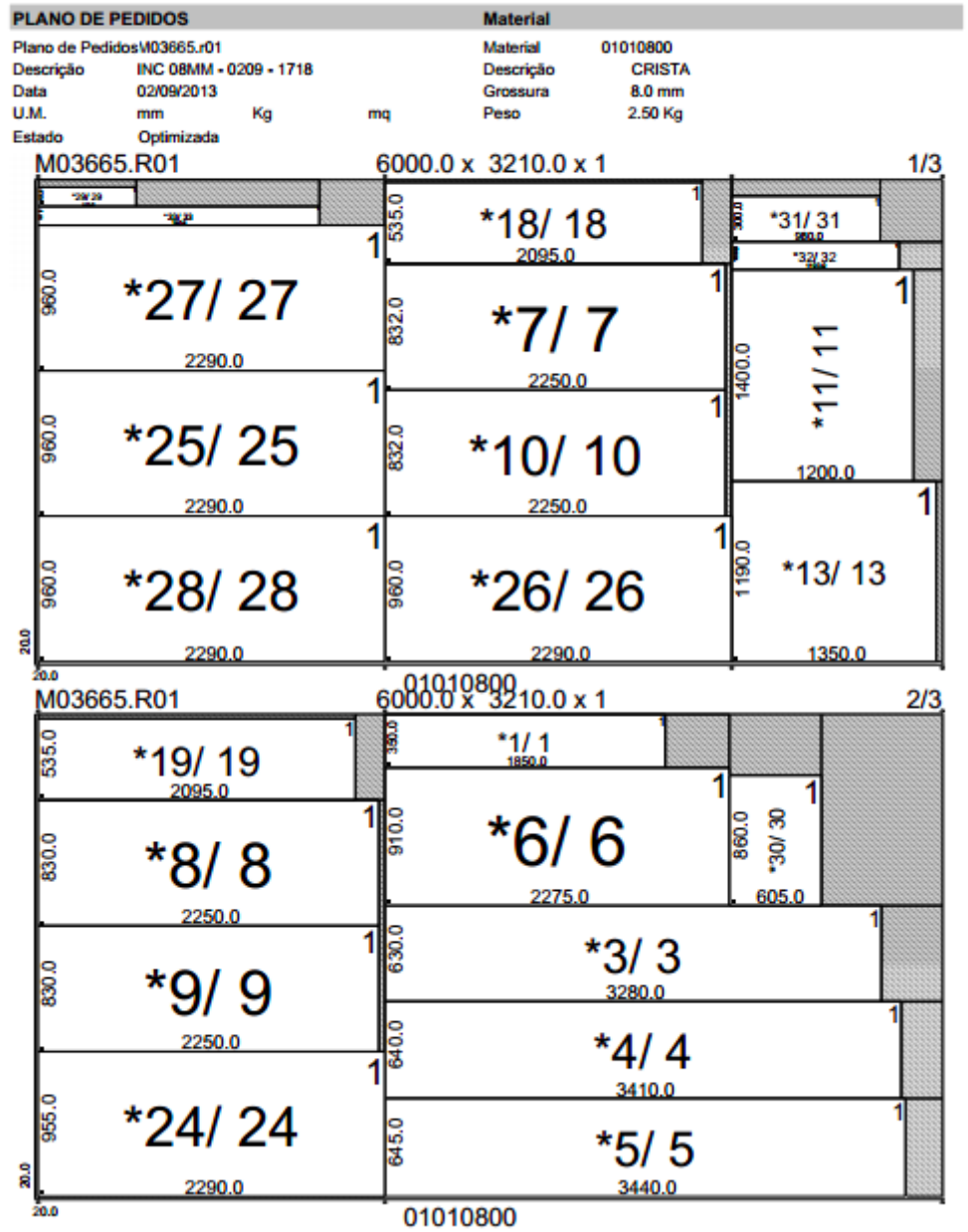
ANEXO B – Exemplos de resultados de processamento dos softwares comerciais.

A16 /790	A56/790 1043*323	A54/790 1043*323	A63/790 1043*343	A18 /790	A07 /790	B33/790  783*853	B36/790  783*853	
	A32/790 1043*403	A36/790 1043*403	A47/790 1043*403					
	403 1043	A33/790 1043*403	A12/790 1043*403					A27/790 1043*403
A31 /790	A34/790 1043*403	A14/790 1043*403	A46/790 1043*403	A08 /790	A51 /790	A06/791	A05/791	
	403 1043	A38/790 1043*403	A17/790 1043*403					A19/790 1043*403
	A23/790 1043*403	A22/790 1043*403	A39/790 1043*403					403 1043
A43 /790	A20/790 1043*403	A13/790 1043*403	A25/790 1043*403	A09 /790	A30 /790	813*2196	813*2196	
	403 1043	A15/790 1043*403	A11/790 1043*403					A10/790 1043*403
	403 1043	403 1043	403 1043					403 1043
16	1043	1043	1043	403	403	813	813	20

Stock plate # : 1 to: 1 of: 11 Pane count: 37 Glass: CRISTAL CINZA 8,0MM  
 Stock sheet size: 6000.0 \* 3210.0 Waste: 2.77 % = 0.53 m2  
 Batches: 791 (SG )-L:1,790 (SG )-L:1,

ID-No.	Qty	Total	Width*Height	Order	Item	Customer
A05/791	1	1	813.0 * 2196.0	112383970	1	VIDRACARIA VAZ DE LI
A06/791	1	1	813.0 * 2196.0	112383980	1	VIDRACARIA VAZ DE LI
A07/790	1	1	403.0 * 1043.0	123656080	1	E O DE MELO COMERCIO
A08/790	1	1	403.0 * 1043.0	123656080	2	E O DE MELO COMERCIO
A09/790	1	1	403.0 * 1043.0	123656080	3	E O DE MELO COMERCIO
A10/790	1	1	403.0 * 1043.0	123656080	4	E O DE MELO COMERCIO
A11/790	1	1	403.0 * 1043.0	123656080	5	E O DE MELO COMERCIO
A12/790	1	1	403.0 * 1043.0	123656080	6	E O DE MELO COMERCIO
A13/790	1	1	403.0 * 1043.0	123656080	7	E O DE MELO COMERCIO
A14/790	1	1	403.0 * 1043.0	123656080	8	E O DE MELO COMERCIO
A15/790	1	1	403.0 * 1043.0	123656080	9	E O DE MELO COMERCIO
A16/790	1	1	403.0 * 1043.0	123656080	10	E O DE MELO COMERCIO
A17/790	1	1	403.0 * 1043.0	123656080	11	E O DE MELO COMERCIO
A18/790	1	1	403.0 * 1043.0	123656080	12	E O DE MELO COMERCIO
A19/790	1	1	403.0 * 1043.0	123656080	13	E O DE MELO COMERCIO
A20/790	1	1	403.0 * 1043.0	123656080	14	E O DE MELO COMERCIO
A22/790	1	1	403.0 * 1043.0	123656080	16	E O DE MELO COMERCIO
A23/790	1	1	403.0 * 1043.0	123656080	17	E O DE MELO COMERCIO
A25/790	1	1	403.0 * 1043.0	123656080	19	E O DE MELO COMERCIO
A27/790	1	1	403.0 * 1043.0	123656080	21	E O DE MELO COMERCIO
A30/790	1	1	403.0 * 1043.0	123656080	24	E O DE MELO COMERCIO

Exemplo de resultado apresentado pelo Lisec GPS.opt..



Exemplo de resultado apresentado pelo Opty way.



## APÊNDICES

**APÊNDICE 1** - Trecho do código em Java que gera os cromossomos e formigas aleatoriamente:

```

private ArrayList<Integer> sequenciaAleatoria(int tamanho) {
    ArrayList<Integer> aux = new ArrayList<Integer>();
    ArrayList<Integer> ret = new ArrayList<Integer>();
    int i,j;
    int[] nos;
    nos = new int[tamanho];

    for (j=0;j<tamanho;j++){
        aux.add(new Integer(1+j));
    }
    j=0;
    while (aux.size()>0){
        i = (int) Math.round(Math.random()*aux.size());
        if (i==aux.size())
        {
            i--;
        }
        nos[j++] = ((Integer) aux.get(i)).intValue();
        aux.remove(i);
    }
    for (j=0;j<tamanho;j++){
        ret.add(Integer.valueOf(nos[j]));
    }
    return ret;
}

```

Os códigos originais (LINDEN, 2012) estão disponíveis em:  
 <<http://algoritmosgeneticos.com.br/CromossomoGAOrdem.zip>>. Acesso em: 20 mar. 2014.

**APÊNDICE 2** - Trecho do código em Java que retorna a avaliação dos indivíduos e formigas artificiais, conforme a equação de Falkenauer:

```

public double calculaAvFalk(int nivel) {
    double areaTotal = 0;
    double areaVazia = 0;
    double avaliacao = 0.0;
    double F = 0.0; //conteudo total da chapa
    double C = 0.0; //conteudo possivel
    double res = 0.0; //resultado acumulado
    double aL = 0.0;
    int kv = 2; Chapa ch; Coluna co; Linha li;
    for(int i = 0; i < chapas.size(); i++){
        ch = (Chapa) chapas.get(i);
        areaTotal += (ch.largura * ch.altura);
        areaVazia += ch.areaLivre;
        aL = ch.areaLivre;
        for(int j = 0; j < ch.colunas.size(); j++){
            co = (Coluna) ch.colunas.get(j);
            areaVazia += co.areaLivre;
            aL += co.areaLivre;
            for(int k = 0; k < co.linhas.size(); k++){
                li = (Linha) co.linhas.get(k);
                areaVazia += li.areaLivre;
                aL += li.areaLivre;
            }
        }
        F = ((ch.largura * ch.altura) - aL);
        C = (ch.largura * ch.altura);
        res += Math.pow((F/C), kv);
    }
    double avFalk = res/chapas.size();
    if(areaVazia == 0){
        avaliacao = 0;
    } else{
        avaliacao = (areaVazia / areaTotal) * 100;
    }
    avaliacao = (100 - avaliacao);
    return (1 - avFalk);
}

```