

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE
PRODUÇÃO

APLICAÇÃO DA META-HEURÍSTICA GRASP PARA O PROBLEMA DE
CORTE DE ESTOQUE BIDIMENSIONAL ATRAVÉS DE APLICATIVOS
CAD

DISSERTAÇÃO DE MESTRADO

CLICERES MACK DAL BIANCO

SANTA MARIA, AGOSTO DE 2010

**APLICAÇÃO DA META-HEURÍSTICA GRASP PARA O PROBLEMA DE
CORTE DE ESTOQUE BIDIMENSIONAL ATRAVÉS DE APLICATIVOS
CAD**

Desenvolvida por

CLICERES MACK DAL BIANCO

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Engenharia de Produção, Área de Concentração em Gerência da Produção, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Engenharia de Produção.**

Orientador: Prof. Dr. Alexandre Dias da Silva

SANTA MARIA, AGOSTO DE 2010

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE
PRODUÇÃO

A comissão abaixo assinado, aprova a dissertação

APLICAÇÃO DA META-HEURÍSTICA GRASP PARA O PROBLEMA DE
CORTE DE ESTOQUE BIDIMENSIONAL ATRAVÉS DE APLICATIVOS
CAD

Elaborada por

CLICERES MACK DAL BIANCO

Como requisito parcial para obtenção do grau de mestre em Engenharia de Produção

COMISSÃO EXAMINADORA:

Prof. Dr. Alexandre Dias da Silva – Orientador – UFSM

Prof. Dr. André Zanki Cordenonsi – UFSM

Prof. Dr. Inácio da Fontoura Limberger – UFSM

SANTA MARIA, AGOSTO DE 2010

AGRADECIMENTOS

Ao Professor Alexandre, meu orientador, pela oportunidade, confiança a mim dedicada na elaboração deste trabalho, pelas horas de ensinamentos e sabias explicações e por seu exemplo de profissional.

Ao Francisco, diretor da indústria de móveis Back, por estar sempre disposto a nos receber e pela valiosa contribuição na realização deste trabalho.

Aos professores Dr. André Zanki Cordenonsi e Inácio da Fontoura Limberger pelo auxílio para melhoria deste trabalho como membros da banca examinadora.

Ao Sandro, meu namorado, pelas leituras e sugestões que enriqueceram este trabalho, pela constante paciência, incentivo, motivação e por tantos e tantos finais de semana em casa, durante a redação da dissertação.

À Prof. Mari Mariza Barthel, diretora da Escola Felisberto de Carvalho, pela amizade e pelas liberações concedidas imprescindíveis na realização deste trabalho.

Ao Prof. Leandro Rosniak Tibola, coordenador do Curso de Ciências da Computação da URI Campus – Frederico Westphalen, pela compreensão durante o período deste trabalho.

Aos professores da Universidade Federal de Santa Maria, funcionários e colegas que direta ou indiretamente contribuíram para a elaboração deste trabalho.

Aos meus pais, pela dedicação incondicional e compreensão nos momentos de ausência.

Aos meus irmãos, Deoclíceo e Letícia pelo carinho e incentivo e ao João Guilherme pela alegria que proporciona nas reuniões familiares.

As minhas primas, Glaucia e Janislene que me acolheram no tempo que morei e nas idas a Santa Maria.

A Capes pelo apoio financeiro.

A Deus por iluminar e guiar os penosos caminhos.

Enfim agradeço a todos tanto do lado profissional quanto pessoal que me auxiliaram nesta etapa da minha vida. Obrigada a todos!

RESUMO

O Problema de Corte de Estoque Bidimensional (PCEB) consiste em cortar um conjunto de objetos disponíveis em estoque para produzir uma quantidade de itens com tamanhos específicos, de modo a otimizar a matéria prima. Tais problemas têm inúmeras aplicações industriais e tem sido amplamente estudados na literatura. Os PCEB são classificados como problemas de otimização combinatória, pois, dependendo da quantidade de itens solicitados, podem ser combinados de inúmeras maneiras dentro de um objeto formando um arranjo de corte. O número de combinações possíveis é, na prática, muito elevado, exigindo que técnicas bem elaboradas sejam desenvolvidas para determinar um arranjo ótimo. Na definição dos arranjos de corte é necessário representar graficamente os componentes do problema e neste caso, as ferramentas empregadas na implementação devem apresentar capacidade de manipulação geométrica. Neste trabalho, o problema combinatório foi abordado a partir da técnica heurística GRASP e como ferramenta de manipulação geométrica empregou-se aplicativos implementados em ambiente de Projetos Auxiliado por Computador (CAD). Além disso, foi proposta uma integração dos arranjos de cortes com sistemas de Manufatura Auxiliador por Computador (CAM). A análise de desempenho do algoritmo desenvolvido considera dados fornecidos por uma indústria moveleira da região de Santa Maria. Os resultados obtidos pelo algoritmo foram comparados aos resultados gerados por *software* especializado, constando-se que a metodologia proposta apresenta um desempenho semelhante, o que demonstra a viabilidade o uso das ferramentas CAD para implementação de soluções PCE em indústria de pequeno e médio porte.

Palavras – Chaves: Problema de corte de estoque bidimensional, técnica GRASP, Sistemas CAD/CAM.

ABSTRACT

The PCE (Cutting Stock Problem) is to cut a set of objects available in stock to produce a quantity of items with specific sizes in order to optimize the raw material. Such problems have numerous industrial applications and has been widely studied in literature. The PCE's are classified as combinatorial optimization problems, because depending on the amount of requested items can be combined in numerous ways within an object forming an arrangement of cut. The number of possible combinations is, in practice, very high, requiring well-developed techniques are developed to determine an optimum arrangement. In defining the arrangements of cut is needed to graph the components of the problem and in this case, the tools used in the implementation must have the ability to manipulate geometric. In this paper, the combinatorial problem was approached from the GRASP heuristic technique and as a tool of manipulation was employed geometric applications deployed in an environment of Computer Aided Design (CAD). Furthermore, we proposed an integration of arrays of cutting systems with computer-aided manufacturing (CAM). The analysis of performance of the developed algorithm considers data provided by a furniture industry in the region of Santa Maria. The results obtained by the algorithm were compared to results generated by specialized software, consisting that the proposed method presents a similar performance, which demonstrates the feasibility of using CAD tools to implement solutions in the industry of PCE small and medium businesses.

Keywords: Cutting Stock Problem, GRASP technique, CAD / CAM systems.

LISTA DE FIGURAS

FIGURA1 - PROBLEMA UNIDIMENSIONAL.....	17
FIGURA 2 - PROBLEMA BIDIMENSIONAL.....	18
FIGURA 3 - PROBLEMA TRIDIMENSIONAL FONTE ARAUJO(2006).	18
FIGURA 5 - (A) CORTE GUILHOTINADO; (B) CORTE NÃO-GUILHOTINADO.	21
FIGURA 6 – ARRANJO DE CORTE ESTAGIADO.....	22
FIGURA 7 - (A) ARRANJO DE CORTE EXATO; (B) ARRANJO DE CORTE NÃO EXATO.....	22
FIGURA 8 – (A) POSICIONAMENTO ORTOGONAL; (B) POSICIONAMENTO NÃO-ORTOGONAL.	23
FIGURA 9 – CRITÉRIO DE SOBREPOSIÇÃO.	24
FIGURA 10 – PSEUDOCÓDIGO GENÉRICO DA META-HEURÍSTICA GRASP.	29
FIGURA 11 – A META-HEURÍSTICA GRASP NA SOLUÇÃO DO PROBLEMA.....	29
FIGURA 12 – FASE DE CONSTRUÇÃO DA GRASP.....	30
FIGURA 13 – PSEUDOCÓDIGO BUSCA_LOCAL.	31
FIGURA 14 – FORMATO TABULEIRO. FONTE: RANGEL E FIGUEIREDO 2008.....	33
FIGURA 15 – SEGUNDO MÉTODO DA BUSCA LOCAL DE ALVAREZ-VALDES ET AL (2005).	34
FIGURA 16 – CORTE HORIZONTAL E VERTICAL SIMULANDO A FAIXA.	35
FIGURA 17 – ESTRUTURA DE DADOS PARA O MODELO B-REP. FONTE CUNHA 2000.	40
FIGURA 18 – EXEMPLO DE MODELAMENTO EM SUPERFÍCIE. FONTE SOUZA 2003.	40
FIGURA 19 – SISTEMAS CAD/CAM OPERANDO DE FORMA INTEGRADA.	42
FIGURA 20 – TÉCNICA <i>BOTTOM-LEFT</i>	47
FIGURA 21 – EXEMPLO DA ESTRUTURA SOLUÇÃO PARA O PCEG.	48
FIGURA 22 – PROCESSO PARA GERAÇÃO DE FAIXAS HORIZONTAIS.	51
FIGURA 23 – REPRESENTAÇÃO GRÁFICA DAS PEÇAS.....	53
FIGURA 24 – CAMADA DEMANDA E CAMADA ARRANJO 1.....	54
FIGURA 25 – (A) PEÇA A SER POSICIONADA E O SEU PONTO DE REFERÊNCIAS; (B) CHAPA E O PONTO DESTINO; (C) A CHAPA COM A PEÇA E O PRÓXIMO PONTO DE DESTINO.....	55

FIGURA 26 – (A) PEÇA QUE SERÁ POSICIONADA NA CHAPA; (B) PEÇA ROTACIONADA 90° (C) POSICIONAMENTO DA PEÇA NA CHAPA	56
FIGURA 27 – DEMONSTRAÇÃO DOS CORTES PARA A TÉCNICA ARTESANAL.	57
FIGURA 28 – DADOS DE UMA ENTIDADE CAD ACESSADOS ATRAVÉS DA ROTINA PROPRI_PEÇA.	58
FIGURA 29 – ALGORITMO PARA CALCULAR AS DIMENSÕES DAS PEÇAS.	59
FIGURA 30 – FUNÇÃO EMPREGADA PARA GERAR AS CHAPAS.	60
FIGURA 31 – EXEMPLO DE CHAPAS QUE RESTRINGEM A ROTAÇÃO.	60
FIGURA 32- FUNÇÃO VERIFICA RESTRIÇÃO QUANTO AO VEIO.	60
FIGURA 33 – PROCEDIMENTO CONSTRUTIVO DO ALGORITMO GRASP-PCEB.	62
FIGURA 34 – FUNÇÃO DE MELHORAMENTO.	63
FIGURA 35 – ALGORITMO GRASP –PCEB.....	63
FIGURA 36 – MÓVEL ARMÁRIO.....	66
FIGURA 38 – ARRANJOS DE CORTE GERADOS PELO PROGRAMA DE REFERÊNCIA	68
FIGURA 39 – ARRANJOS DE CORTE PELO MÉTODO ARTESANAL	69
FIGURA 40 – REPRESENTAÇÃO 3D DO MÓVEL AÉREO.....	69
FIGURA 43 - ARRANJO GERADO PELO PROGRAMA GRASP- PCEB	71
FIGURA 44 – ARRANJO GERADO PELO PROGRAMA DE REFERÊNCIA	72
FIGURA 45 – ARRANJO GERADO PELA TÉCNICA ARTESANAL.....	72
FIGURA 47 – ARRANJOS COM RESTRIÇÃO, GERADO PELO PROGRAMA DE REFERÊNCIA.	73
FIGURA 48 – REPRESENTAÇÃO DO MÓVEL BALCÃO.....	74
FIGURA 49 – ARRANJO DE CORTE (A)GRASP-PCEB; (B) DE REFERÊNCIA.	74
5.4 PROGRAMA CNC.....	75
FIGURAS 51 – COMPONENTES DA MÁQUINA SECCIONADORA	76
FIGURA 52 – ARRANJO DE CORTE MODELO.	77
FIGURA 53 – PROGRAMA CN PARA O ARRANJO MODELO.FIGURA 54 – PROCESSO DE CORTE EMPREGADO PELO PROGRAMA CNC.	78
FIGURA 54 – PROCESSO DE CORTE EMPREGADO PELO PROGRAMA CNC.	79

LISTA DE TABELAS

TABELA 1 - CLASSIFICAÇÃO PARA PCE DE ACORDO COM DYCKHOFF (1990).....	17
TABELA 2 – COMANDOS CNC	44
TABELA 3 – RELATÓRIO DE CORTE.....	53
TABELA 4 – MODELO DE RELATÓRIO DE CORTE.	536
TABELA 4 – DADOS DAS INSTÂNCIAS B2P, A3P E R5P.....	64
TABELA 5 – RESULTADO PARA INSTÂNCIA R5P.....	66
TABELA 6 – RESULTADOS DA INSTÂNCIA A3P.....	71
TABELA 7 – RESULTADOS DA INSTÂNCIA B2P.....	74

LISTA DE SIGLAS

CAD	Projeto Auxiliado por Computador
CAM	Manufatura Auxiliada por Computador
CN	Comando Numérico
CNC	Comando Numérico Computadorizado
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
ISO	International Standard Organization
LRC	Lista Restrita de Candidatos
PCE	Problema de Corte e Empacotamento
PCEB	Problema de Corte de Estoque Bidimensional

SUMÁRIO

1. INTRODUÇÃO	12
1. 1 PROBLEMA DE CORTE.....	12
1. 2. OBJETIVOS	13
2. REVISÃO DA LITERATURA.....	14
2.1 AMBIENTE INDUSTRIAL	14
2.3 TIPOLOGIAS DO PROBLEMA DE CORTE.....	16
2.3.1 Critério de classificação.....	16
2.3.2 Tipo de corte: guilhotinado e não-guilhotinado	20
2.3.3 Arranjo de Corte	21
2.3.4 Orientação dos itens.....	21
2.3.5 Número de estágio.....	22
2.3.6 Desperdício interno e externo.....	23
2.3.7 Posicionamento Ortogonal e não ortogonal.....	23
2.4 O PROBLEMA DE CORTE DE ESTOQUE BIDIMENSIONAL	23
2.5 TÉCNICAS DE RESOLUÇÃO.....	25
2.6 EXEMPLOS DE META-HEURÍSTICAS	27
2.7 GRASP.....	28
2.7.1 Fases da GRASP.....	30
2.8 TRABALHOS RELACIONADOS.....	32
2.9 MODELAGEM GEOMÉTRICA.....	36
2.9.1 Modelagem de Sólidos.....	38
2.9.2 Modelagem de Superfície.....	40
2.9.3 Modelagem Híbrida.....	41
2.10 SISTEMAS CAD/CAM	41
2.11 COMANDO NUMÉRICO COMPUTADORIZADO (CNC).....	43
2.12 APLICATIVOS CAD.....	45
3. METODOLOGIA.....	46
3.1 CRITÉRIOS PARA ORDENAÇÃO DAS PEÇAS	46
3.2 TÉCNICA <i>BOTTOM-LEFT</i>	47
3.3 ESTRUTURAS PARA A CONSTRUÇÃO DO ALGORITMO.....	47
3.4 APLICAÇÃO DA META-HEURÍSTICA GRASP AO PCEB	49
3.4.1 Fase de Construção.....	50
3.4.2 Fase de Melhoramento	51
3.5 REPRESENTAÇÃO GEOMÉTRICA	52
3.5.1 Posicionamento das Peças.....	54
3.5.2 Orientação das peças	55
3.6 TÉCNICA ARTESANAL.....	56
4. DESENVOLVIMENTO	58
4.1 IMPLEMENTAÇÃO DO ALGORITMO GRASP – PCEB	59
5. RESULTADOS	64
5.1 INSTÂNCIA 1.....	65
5.2 INSTÂNCIA 2.....	69
5.3 INSTÂNCIA 3.....	74
6. CONCLUSÃO.....	80
7 REFERÊNCIAS BIBLIOGRÁFICAS	82

1. INTRODUÇÃO

1.1 Problema de Corte

Atualmente as indústrias enfrentam uma difícil tarefa ligada às tomadas de decisões que são vitais à sua sobrevivência no mercado. Muitas dessas decisões estão relacionadas às atividades de produção e logística. Essas ações visam reduzir os custos dessas atividades, implicando em economia de recursos financeiros, as quais favorecem a competitividade.

Em algumas indústrias, como por exemplo, de calçados, móveis, confecções, metal-mecânica e construção civil, uma maneira de reduzir os custos, em suas atividades de produção, pode ser obtida durante o processo de corte da matéria-prima. Nestas indústrias, a matéria-prima se apresenta no formato de barras de aço, bobinas de papel de alumínio e de aço, placas metálicas, de madeira, de vidro e de couro, que necessitam ser cortadas em itens menores (peças), para atender uma determinada demanda. Devido ao formato padrão da matéria-prima e a forma pela qual é processada, pode causar desperdício resultando em aumento substancial de custos e, conseqüentemente, diminuindo a competitividade. Na literatura, esse processo é conhecido como Problema de Corte e Empacotamento (PCE).

Os PCE's pertencem a classe de problemas Não-Polinomial-Difícil (NP-difícil) significando que não existe algoritmo exato que resolva este problema em tempo polinomial. Esta inviabilidade é atribuída á explosão combinatória de arranjos possíveis quando o objetivo é a determinação de um arranjo de corte ótimo. Neste caso, o uso de técnicas exatas para otimização necessitam de um longo tempo de processamento computacional e não propiciam resultados práticos.

Uma alternativa para resolver os problemas da classe NP-Difícil é o emprego de técnicas heurísticas, que não garantem a obtenção da solução ótima para o problema, mas asseguram soluções sub-ótimas e em menor tempo computacional, quando comparado à utilização de métodos exatos (NORONHA, 2005).

Os PCE's se caracterizam pela facilidade que são representados através de modelos matemáticos, entretanto, são problemas difíceis de serem solucionados devido à componente geométrica que envolve o problema. Deste modo, as ferramentas empregados na implementação

devem permitir: representação gráfica dos componentes do problema; desenho através de comandos; processamento de arquivos; leitura das informações do desenho; realização de operações matemáticas e verificação das sobreposições dos itens.

Ainda de acordo com Gomes, (2008) o desenvolvimento de soluções competitivas para os PCE's torna quase que obrigatória a utilização de uma eficiente biblioteca geométrica. Neste contexto, os sistemas de Projetos e Manufaturas Auxiliados por Computador (CAD/CAM) constituem uma opção válida para contornar essas dificuldades impostas pela componente geométrica do problema.

O uso de sistemas CAD no desenvolvimento de soluções vem suprir a necessidade de bibliotecas geométricas minimizando a complexidade do problema, pois apresentam comandos práticos facilitando o desenvolvimento de soluções. Enquanto, o emprego dos sistemas CAM oferecem suporte para converter as soluções em Comando Numérico Computadorizado (CNC) gerando instruções para a máquina de corte.

1. 2. Objetivos

a) Geral

Este trabalho teve como objetivo principal desenvolver uma metodologia para resolução do problema de corte bidimensional através da implementação de aplicativos em ambiente CAD.

b) Específicos

- Desenvolvimento de um aplicativo visando gerar os arranjos de corte empregando técnica GRASP e recursos CAD.
- Testar a viabilidade do aplicativo como um módulo de sistema CAD/CAM dedicado para a indústria moveleira.
- Análise dos resultados obtidos verificando a eficiência do programa desenvolvido em relação à qualidade da solução final.
- Apresentação de uma técnica para converter os padrões de corte para códigos CNC objetivando instruir máquinas programáveis.

2. REVISÃO DA LITERATURA

As indústrias do ramo de confecções, calçados, móveis e metal-mecânico têm seus processos de produção ligados ao corte de matéria-prima. Em geral a matéria-prima se apresenta em tamanhos padronizados e estocados para ser cortada em itens menores, de tamanhos variados e não padronizados, a fim de atender à demanda interna e/ou externa. Este processo é conhecido na literatura como Problemas de Cortes e Empacotamentos ou *Cutting and Packing Problems* na versão inglesa.

Nestas indústrias o planejamento do corte da matéria-prima é essencial a fim de minimizar o desperdício de material e garantir a competitividade das indústrias no seu setor de atividades.

2.1 Ambiente Industrial

As indústrias que apresentam PCE, apesar de partilharem o mesmo problema base, estas indústrias apresentam algumas características específicas que diferenciam os respectivos problemas, assim torna-se natural o desenvolvimento de aplicações de corte, apropriadas e específicas para cada tipo de indústria (GOMES et al., 2005).

Cherri (2009) constata que são várias as situações em que surgem os PCE, cada um deles com suas especificidades, restrições e objetivos definidos pelas exigências práticas impostas em cada ambiente em que estes problemas aparecem. Neste contexto, torna-se importante descrever as situações existentes e apresentar as principais características das indústrias de: confecções, metal-mecânica, calçados e de móveis.

A indústria de confecções apesar de ser uma indústria baseada em mão-de-obra intensiva encontra-se numa fase mais avançada em termos de aplicações e ferramentas de corte. De acordo com Hopper (2003) esta indústria é considerada a pioneira em relação a pesquisa e desenvolvimento de aplicações informatizadas para o apoio ao corte de itens com formato regular e irregulares.

Segundo Burke et al. (2004), o processo corte é uma das áreas mais importante na indústria de confecções, principalmente devido ao elevado custo das matérias-primas, e destas

principalmente o tecido, que na generalidade das indústrias têxteis se aproxima dos 50% dos custos totais de produção. Existem diversas aplicações e módulos específicos para a obtenção de padrões de corte de boa qualidade para a indústria de confecções, a grande maioria delas está integrada em sistemas que permitem controlar automaticamente as máquinas de corte. No entanto, em todas estas aplicações existe sempre a possibilidade de melhorar a qualidade dos padrões de corte.

O problema de corte, que surge na indústria metal-mecânica é o que de base apresenta menos limitações. As placas metálicas são retangulares, com dimensões finitas, homogêneas e sem defeitos, sendo representadas por um simples retângulo. Os itens não apresentam restrição quanto a orientação.

A indústria metal-mecânica, assim como, a indústria de confecções, apresentam maior número de ferramentas automáticas para o corte de itens. Segundo Oliveira (2005), dentro deste setor industrial, o subsetor da construção naval foi o precursor no desenvolvimento das soluções de posicionamento automático, sendo seguido nos tempos mais recentes pela indústria automobilística e pela indústria aeronáutica. O desenvolvimento neste setor industrial deve-se aos investimentos dos estaleiros navais e das multinacionais de construção de automóveis e aviões.

Entretanto, a indústria de calçados apresenta uma série de particularidades que dificultam a utilização de ferramentas automatizadas no processo de corte. As dificuldades específicas desta indústria devem-se principalmente, a pele, sua principal matéria-prima. Sendo um produto de origem natural, a pele caracteriza-se pela não homogeneidade, a possibilidade da existência de defeitos e formas altamente irregulares, que naturalmente obrigam o planejamento de padrões de corte para cada pele individualmente (FERNANDES, 2000).

Em relação as indústria do setor moveleiro a grande maioria dessas empresas são caracterizadas pela reunião de diversos processos de produção, envolvendo diferentes matérias primas e uma diversidade de produtos finais. Aspectos técnicos e mercadológicos, no entanto, levam as empresas a se especializarem em uma ou duas linhas de produção (por exemplo, de cozinha, dormitórios, banheiro, estofados) (CAVALI, 2004).

Uma característica marcante da indústria moveleira no Brasil é a dispersão geográfica. No entanto as indústrias apresentam-se agrupadas em pólos regionais. Estes pólos estão localizados basicamente na região Sul, com 45,95% da produção nacional, e na região Sudeste, com 39,57%. As regiões Nordeste, Norte e Centro-Oeste apresentam os índices de produção 6,92%, 3,97% e 3,58% respectivamente (ABIMOVEL, 2010).

De acordo com Mosquera (2008) a indústria de móveis caracteriza-se por apresentar elevados níveis de desperdício de matéria-prima, entre 30 e 40%. Existem várias razões para estes

elevados valores de desperdício, das quais merecem particular destaque: a pouca precisão do processo de corte e o grande tamanho das peças comparativamente ao tamanho das placas.

Na grande maioria das indústrias de mobiliário em madeira, o planejamento dos arranjos de corte é totalmente manual, realizado pelo próprio operador da máquina de corte, e ainda a existência de um veio na madeira limita as possibilidades de combinações e a rotações das peças (GOMES, 2008).

2.3 Tipologias do Problema de Corte

Problemas de corte e empacotamento (PCE) é a denominação de uma classe de problemas de otimização combinatória aplicado a problemas que se enquadrem na seguinte situação: uma ou mais unidades grandes, que podem representar um determinado material ou espaço, devem ser divididas em unidades pequenas. Em geral, o objetivo considerado refere-se à minimização do desperdício, ou seja, a quantidade do material ou espaço não utilizado das unidades grandes (TEMPONI et al., 2007).

No caso dos Problemas de Empacotamento (PE), o objetivo geral consiste em empacotar os itens dentro dos objetos. Por outro lado os Problemas de Corte (PC) visam determinar uma forma para cortar os objetos, produzindo itens.

2.3.1 Critério de classificação

Embora os objetivos dos problemas de corte e empacotamento sejam equivalentes, as diferenças ocorrem quando considera-se as diferentes restrições de cada caso. De acordo Wäscher et al. (2007), os Problemas de Corte e Empacotamento podem ser divididos em cinco subproblemas, que possuem diferentes objetivos:

- Problema de seleção de objetos: quando determinados objetos possuem características diferentes, como dimensões, custo, material;
- Problema de seleção de itens: quando determinados itens possuem utilidades diferentes, tendo prioridade em relação aos outros;
- Problema de agrupamento de itens: quando um determinado conjunto de itens não pode ficar junto com outro, como, por exemplo, produtos químicos e produtos alimentícios;

- Problema de alocação de itens em objetos: quando determinados itens podem ser alocados somente em determinados objetos;
- Problema de *layout*: quando os itens devem ser dispostos nos objetos, respeitando-se as condições geométricas.

Dyckhoff (1990) apresentou uma tipologia para problemas de corte e empacotamento, o na qual propõe quatro critérios para classificar os problemas de corte e empacotamento: dimensionalidade, tipos de atribuição, tipo do objeto e tipo do item (Tabela 1).

Tabela 1 - Classificação para PCE de acordo com Dyckhoff (1990).

1. Dimensionalidade	2. Tipo de atribuição	3. Tipo dos objetos	4. Tipo dos itens:
(1) unidimensional	(B) todos objetos e uma seleção de itens	(O) um objeto;	(F) poucos itens (de formas diferentes);
(2) bidimensional	(V) todos itens e uma seleção de objeto	(I) objetos idênticos;	(M) muitos itens (muito heterogêneo)
(3) tridimensional	-----	(D) objetos diferentes;	(R) muitos itens (pouco heterogêneo)
(N) N dimensões, com $N > 3$	-----	-----	(C) itens iguais.

Cada tipo de problema de corte é definido como sendo uma quádrupla $\alpha/\beta/\gamma/\delta$, na qual α é a dimensionalidade; β é o tipo de atribuição; γ é o tipo de objeto; e δ é o tipo de item. Por exemplo, um problema de corte cujo objetivo seja determinar a melhor maneira de cortar barras idênticas de 12 metros de comprimento para produzir barras menores de 3, 5 e 7 metros, é classificado, segundo Dyckhoff (1990), pela quádrupla 1/V/I/F.

O número de dimensões é o primeiro critério considerado, ou seja, as dimensões relevantes para a resolução do problema, sendo distinguidas quatro situações:

Unidimensionais (1D) - O problema é unidimensional quando apenas uma das dimensões é relevante no processo de corte. Ocorre, por exemplo, no processo de corte de barras de aço com a mesma seção transversal, em bobinas de papel. Exemplo: Corte de vigas (madeira, aço, ...) (Figura 1).

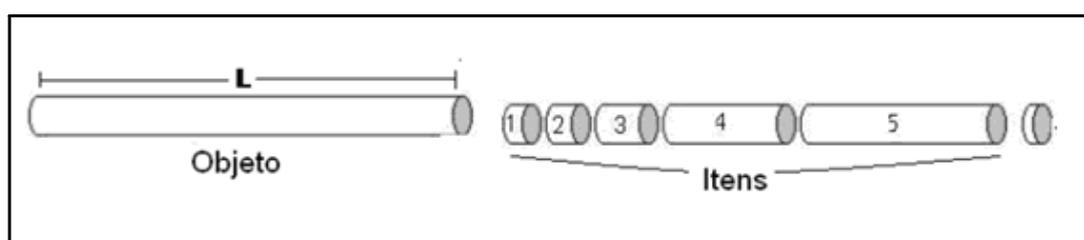


Figura1 - Problema unidimensional.

Bidimensionais (2D) - O problema é bidimensional quando duas dimensões, largura (L) e comprimento (C), são relevantes no processo de corte. Este problema ocorre, por exemplo, no corte de chapas retangulares de madeira, de chapas de aço, de placas de vidro, *etc.* (Figura 2).

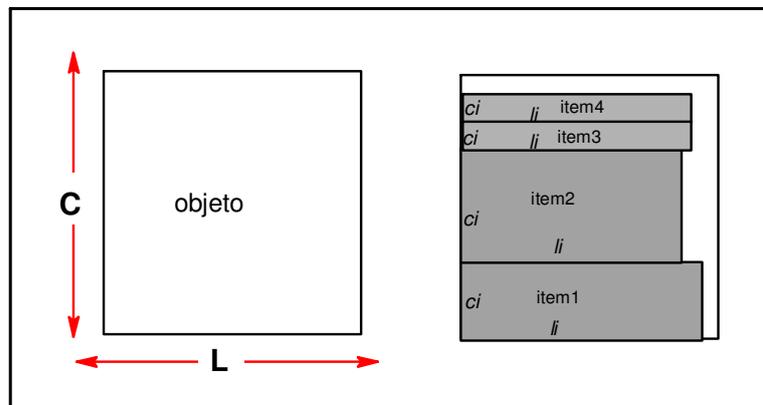


Figura 2 - Problema bidimensional.

Tridimensionais (3D) - O problema é tridimensional quando três dimensões são relevantes no processo de corte (Figura 3). Exemplo: corte de blocos de espumas para fabricação de colchões, travesseiros, *etc.* O maior número de aplicações práticas ocorre na solução para o problema de empacotamento. Por exemplo, o carregamento de contêiner.

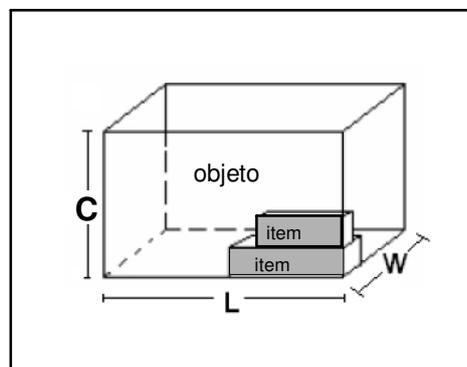


Figura 3 - Problema tridimensional Fonte Araujo(2006).

N-dimensionais - O problema é N-dimensional quando são relevantes mais de três dimensões. Por exemplo, o investimento no mercado financeiro em vários períodos

O **Tipo de atribuições** é o segundo critério e diz respeito à seleção dos objetos e itens. No caso do critério de todos os itens e uma seleção de objeto, onde a quantidade de objetos existentes em estoque não é suficiente para atender todos os itens demandados e, com isto, alguns itens não são selecionados. Exemplos: o problema da mochila. No entanto, a atribuição: uma seleção de objetos e todos os itens, a quantidade de objetos existentes em estoque é suficiente

para atender todos os itens demandados e, assim alguns objetos não são selecionados. Exemplos: problema de corte de estoque.

O terceiro critério está relacionado com o tipo do objeto.

Apenas um objeto - vários objetos todos com o mesmo formato e tamanho onde somente um objeto é considerado. Exemplo malha de tecido.

Objetos diferentes - os objetos apresentam formatos e tamanhos variados. Exemplo: chapas retangulares (usadas no aproveitamento de retalhos em indústria de móveis) e fibra de vidro (usada na fabricação de circuito impresso).

As características dos itens é o quarto critério da tipologia e, são analisadas as seguintes situações:

Poucos itens de formas diferentes - nesta situação, os itens que compõem o problema são todos heterogêneos e se apresentam em pequena quantidade.

Muito heterogêneo – nesta situação, os itens apresentam forma e tamanho diferentes, dificilmente são agrupados.

Pouco heterogêneo - neste caso, os itens permitem o agrupamento em classes de itens idênticos.

Iguais: situação onde os itens apresentam forma e tamanho idênticos.

Com o aumento das pesquisas ligadas ao PCE as variâncias deste problema cresceram e esta tipologia se tornou incompleta. Uma extensão desta tipologia foi proposta em Wäscher et al. (2007), que descreveram o problema de maneira precisa e abrangente. Esta nova tipologia é composta de cinco critérios:

Dimensionalidade: os problemas são divididos, segundo esse critério, em unidimensional, bidimensional e tridimensional.

Tipo de Atribuição: os problemas, de acordo com este critério, podem ter objetivos de:

– Maximizar a Saída: neste caso, todos os objetos disponíveis não são suficientes para alocar todos os itens. Assim, será necessário maximizar a quantidade de itens que é possível produzir a partir de determinada matéria-prima.

– Minimizar a Entrada: neste caso, os objetos disponíveis são suficientes para alocar todos os itens. Assim, é necessário alocá-los buscando minimizar um “valor”, que pode ser, por exemplo, o custo ou a quantidade de material desperdiçado.

Tipo dos Itens: idênticos; pouco heterogêneos; muito heterogêneos.

Tipo de Objetos: um único objeto com todas as dimensões fixas ou com uma ou mais dimensões variáveis; muitos objetos: idênticos; pouco heterogêneos; muito heterogêneos.

Forma dos Itens: regulares; irregulares.

A estrutura da classe dos PCE proposta por Wäscher et al. (2007) é feita em 3 etapas: na primeira etapa, os critérios “tipo de atribuição” e “tipo de itens” são combinados para definir a estrutura básica do PCE; na segunda etapa, o critério “tipo de objetos” é combinado com os problemas básicos para definir a estrutura intermediária; finalmente, na terceira etapa, são adicionados os critérios “dimensionalidade” e “forma dos itens” à estrutura dos problemas. A estrutura básica dos PCE é composta por seis categorias de problemas (Figura 4).

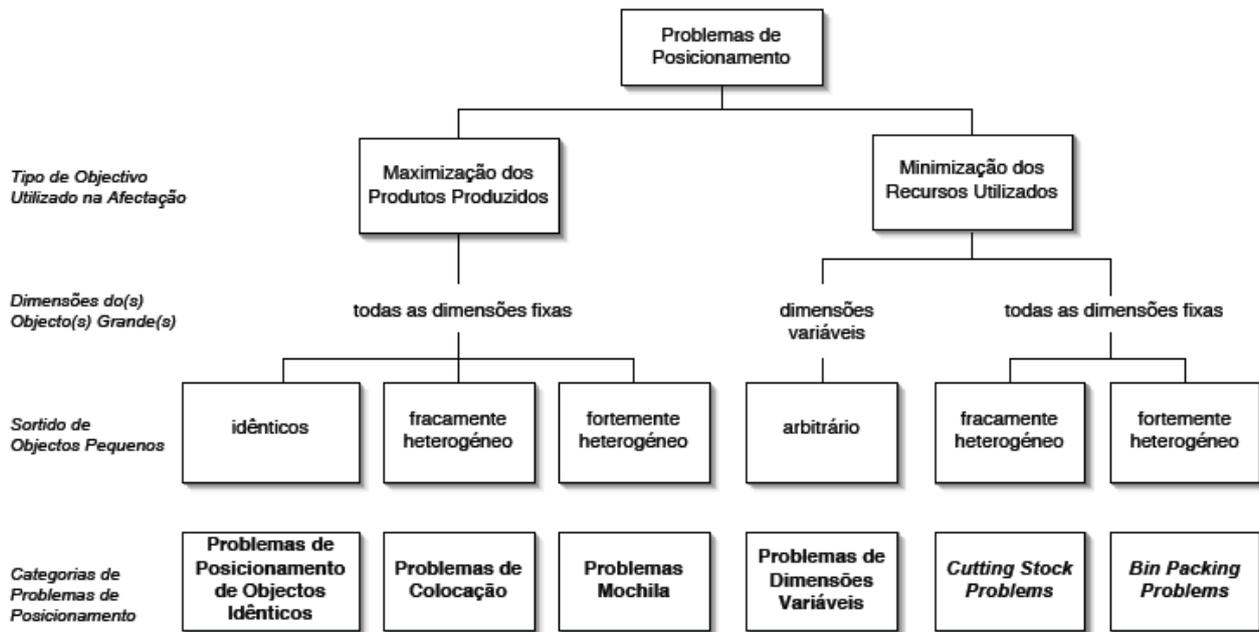


Figura 4 - Categorias de Problemas de corte e empacotamento (Fonte TEMPONI et al., 2007).

O presente trabalho, descreverá o subproblema Cutting Stock Problem, optou-se por chamar de Problema de Corte de Estoque Bidimensional (PCEB). Sendo considerado somente objetos idênticos e itens retangulares, detalhado seções 3.4.

2.3.2 Tipo de corte: guilhotinado e não-guilhotinado

Segundo Morabito e Arenales (1998), quanto ao problema de corte, podem ocorrer dois tipos: o corte guilhotinado e o corte não-guilhotinado. No corte guilhotinado, cada corte é feito em toda a extensão da chapa retangular, gerando assim duas chapas retangulares de menor tamanho do que a original. Geralmente, esse tipo de corte é chamado de guilhotinado de dois estágios, pois o corte é feito primeiro com a guilhotina na vertical, e depois rotacional a guilhotina 90° e faz o corte na horizontal (Figura 5a). No corte não-guilhotinado cada corte é feito

exatamente sobre o tamanho da peça, sem haver descaracterização da chapa retangular, como acontece no corte guilhotina (Figura 5b).

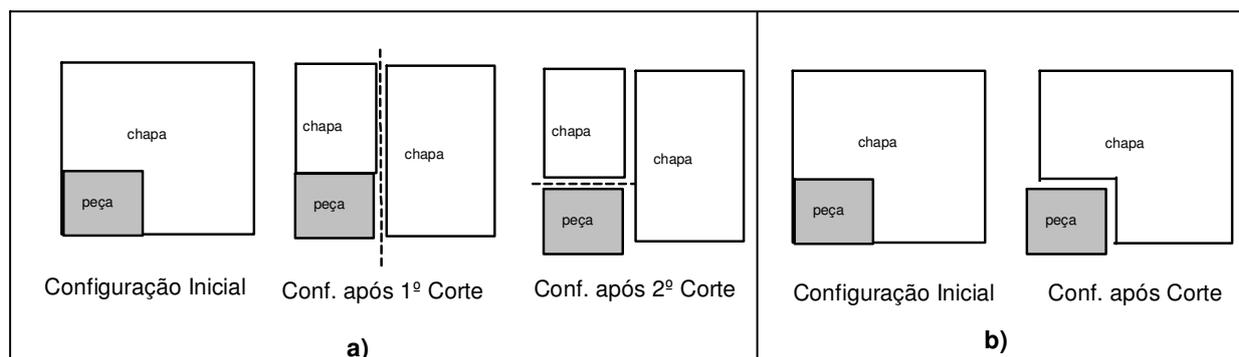


Figura 5 - (a) corte guilhotinado; (b) corte não-guilhotinado.

2.3.3 Arranjo de Corte

De acordo com Alvarez-Valdes et al. (2005) pode-se definir arranjo de corte como sendo a melhor maneira como os itens serão agrupados dentro de um objeto, para serem posteriormente cortados conforme demanda, ou seja, é a solução encontrada a partir dos dados de um problema que pode decorrer de um software ou encontrada de forma empírica.

Pileggi et al. (2007) definem arranjo de corte como o arranjo físico (geométrico) das peças menores dentro das maiores, definindo a seqüência que os itens serão processados a fim de, por exemplo, minimizar o número de sobras. Os arranjos de corte, além de melhorarem o aproveitamento do material, contribuem para a “padronização” da atividade do corte, pois, será realizada segundo “esquemas” preestabelecidos.

2.3.4 Orientação dos itens

O processo produtivo industrial, em alguns casos impõe restrições, uma destas restrições está relacionada com a rotação dos itens, ou seja, um determinado tipo de item deve ser alocado no objeto seguindo a mesma orientação. Esta restrição também é denominada de orientação fixa.

As implementações que trabalham com rotações de itens tornam-se mais complexas, pois aumenta consideravelmente o número de combinações possíveis dos itens (OLIVEIRA, 1995). Com o objetivo de reduzir o número de combinações possíveis, alguns autores consideram apenas rotações ortogonais de 90° (VELASCO, 2008).

2.3.5 Número de estágio

O número de estágios de um arranjo de corte mede a quantidade de mudanças na direção do corte, nas indústrias uma limitação freqüentemente imposta é de estágios de um arranjo de corte (MOSQUERA, 2008).

A Figura 6 ilustra um arranjo de corte, onde são necessárias duas mudanças de direção do corte, caracterizando um arranjo de corte em três estágios. No primeiro estágio, o corte guilhotinado longitudinal resulta em um conjunto de faixas. No segundo estágio, são feitos os cortes guilhotinados transversais em cada faixa. A cada mudança na direção do corte (rotação 90°) para a realização do arranjo de corte, acrescenta-se uma unidade ao número de estágios. Os cortes de dois estágios consecutivos são sempre perpendiculares entre si.

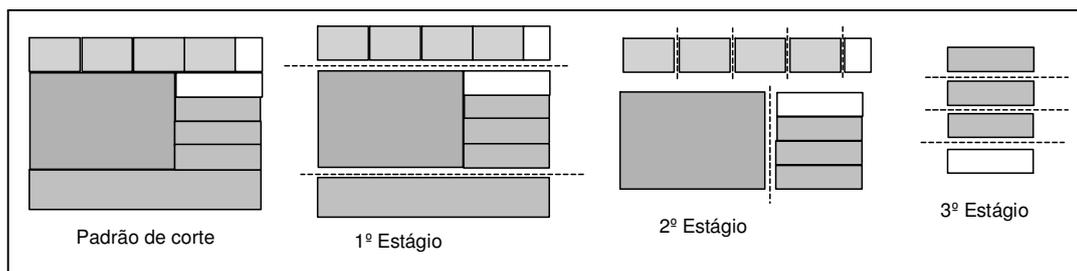


Figura 6 – Arranjo de corte estagiado.

Os estágios são classificados como: exato ou não exato. Um arranjo de corte é considerado exato quando nenhum item precisar de ajuste (apara), conforme apresentado na Figura 7a, caso contrário, este arranjo é chamado de não-exato este corte é demonstrado na Figura 7b.

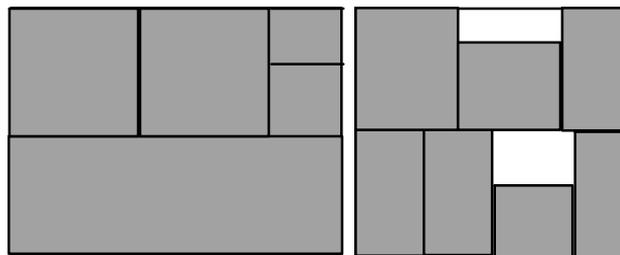


Figura 7 - (a) arranjo de corte exato; (b) arranjo de corte não exato.

2.3.6 Desperdício interno e externo

Conforme as peças vão sendo combinadas no objeto, entre elas pode haver um espaço não aproveitado que chamamos de desperdício interno. Isso ocorre em função das dimensões das peças serem diferentes. Caso as duas peças tenham dimensões iguais, o desperdício interno não existe. O desperdício externo ocorre quando o espaço está além do conjunto de peças projetadas (ARAUJO, 2006).

2.3.7 Posicionamento Ortogonal e não ortogonal

Burke et al. (2004), apresentam uma restrição relacionada à maneira que os itens são alocados dentro dos objetos. Quando os itens retangulares são alocados dentro de objetos também retangulares, de tal forma que os lados dos itens sejam paralelos ou ortogonais aos lados dos objetos, o problema é dito ortogonal, caso contrário, o problema é considerado não-ortogonal (Figura 8). O posicionamento ortogonal é requisito para implementar o padrão guilhotina.

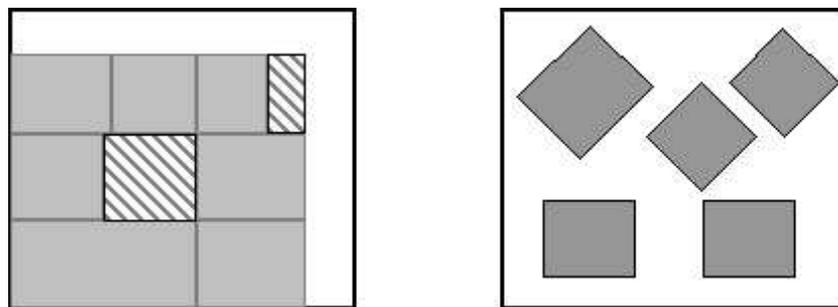


Figura 8 – (a) posicionamento ortogonal; (b) posicionamento não-ortogonal.

2.4 O Problema de Corte de Estoque Bidimensional

A classe de problemas de corte de estoque bidimensional é categoria de problema inserida na classe de problema de corte de estoque. Esta categoria de problema foi objeto de estudo deste trabalho.

O Problema do Corte de Estoque Bidimensional (PCEB) apresenta como principal característica a existência em estoque de uma quantidade suficiente de objetos retangulares de comprimento L e largura W para produzir um conjunto de itens retangulares menores de comprimento l_i , largura w_i (GOLFETO et al., 2008).

Nos problemas de corte de estoque bidimensionais duas dimensões, largura e comprimento, são considerados relevantes. Este problema ocorre em processos produtivos onde as matérias primas apresentam formatos retangulares e necessitam serem cortados itens também de formato retangulares para a obtenção de produtos finais. Esta operação geralmente implica em perda de matéria-prima, com conseqüente aumento nos custos comprometendo a competitividade de diversos setores industriais.

Segundo Suliman (2005), os principais objetivos do problema de corte de estoque são minimizar o número de objetos de estoque necessários para produzir as unidades demandadas e minimizar a perda de material ao arranjar geometricamente as unidades demandadas dentro das unidades de estoque.

Em Lodi et al. (2004) na resolução do problema de corte de estoque guilhotinado algumas condições devem ser respeitadas:

- cada item será alocado exatamente uma única vez;
- o somatório da largura dos itens em cada faixa não irá ultrapassar o limite imposto pela largura do objeto;
- cada faixa será alocada apenas uma vez em algum objeto;
- a condição de que a soma dos comprimentos das faixas alocadas em um determinado objeto não ultrapasse o comprimento do objeto;
- os itens não deverão se sobrepor, para evitar esta sobreposição de itens, ao se decidir arranjar um item do tipo i na posição (p,q) , não se pode colocar outro item na posição (r,s) (Figura 9).

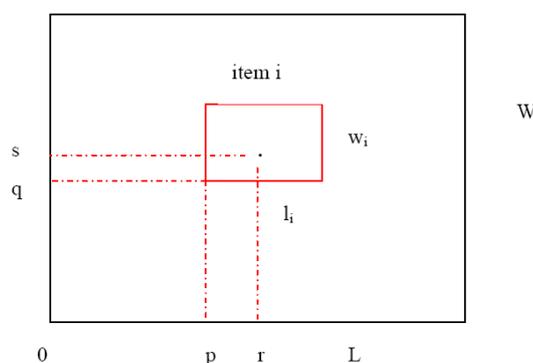


Figura 9 – Critério de sobreposição.

2.5 Técnicas de Resolução

Os PCE são classificados como problemas combinatórios que, do ponto de vista da sua resolução, são classificados como Não-Polinomial-Difícil ou NP-difícil (OLIVEIRA, 1995). Isso significa que não existe algoritmo exato que resolva este problema em tempo polinomial. Esta inviabilidade é atribuída á explosão combinatória de arranjos possíveis quando o objetivo é a determinação de um arranjo ótimo, ou seja, as técnicas exatas de otimização requerem um longo tempo computacional de processamento e não propiciam resultados práticos.

Devido à alta complexidade de resolução de muitos e importantes problemas combinatórios, começaram a surgir algoritmos que encontram a soluções em tempo computacional razoável. Tais algoritmos não garantem encontrar a melhor solução e, às vezes, nem mesmo garantem encontrar uma solução factível (OSMAN, 2002). Para resolver problema da classe NP-Difícil são utilizadas técnicas conhecidas como heurísticas.

Estas técnicas são alternativas viáveis para tratar PCE, uma vez que possibilitem solução de boa qualidade dentro de um tempo razoável, sendo estes, requisitos suficientes para atender a maioria das aplicações práticas (ARAUJO, 2006). O objetivo das heurísticas é tentar encontrar uma solução “boa” de maneira simples e rápida. Existem muitas classificações para heurísticas, mas Miranda (2005) resume em três categorias: Construtivas, Melhoramento e Meta-Heurísticas.

a) Heurísticas Construtivas

As heurísticas construtivas consistem em desenvolver a solução de forma incremental, ou seja, a solução é criada passo a passo. Um componente é escolhido e inserido no processamento até gerar uma solução completa. O componente escolhido em cada passo é o melhor de acordo com algum critério. As heurísticas construtivas, em geral, encontram uma solução rápida e são algoritmos que criam soluções viáveis. Podem ser utilizadas de maneira isolada, contudo, são frequentemente utilizadas em métodos mais elaborados e em conjunto com outro algoritmo heurístico (MIRANDA, 2005).

As técnicas de encaixe são consideradas exemplos de heurísticas construtivas, as mais conhecidas são *Next-fit*, *First-Fit*, *Best-fit*, *Last-fit*, *Worst-fit* e *Bottom-left* (CONSTANTINO et al., 2002).

b) Heurísticas de Melhoramento

As heurísticas de Melhoramentos constroem uma solução pela exploração no espaço de busca. Partem de uma solução inicial, que pode ser obtida de outra heurística. Um algoritmo de melhoramento define, para cada solução, uma vizinhança. A cada iteração, o algoritmo procura trocar um item que compõem a solução por outro que pertence à vizinhança. Essa troca é avaliada. Se houver melhora na função objetivo, o item torna-se parte da solução corrente e o algoritmo continua sua busca por outra solução na vizinhança. Caso não seja encontrada melhora na função objetivo, a solução corrente é um ótimo local em relação à vizinhança adotada. A Busca Local é um exemplo de heurística de melhoramento (MÜLLER, 1993).

c) Meta-Heurísticas

Da década de 80 surgem as meta-heurísticas que se caracterizam por pesquisas sobre vizinhanças, distinguindo-se do tipo anterior por permitirem movimentos para soluções pior qualidade de forma controlada, ou seja, em algoritmos baseados em heurística puras como por exemplo, Busca Local, as soluções acabam por ficar presas em ótimos locais, uma vez que só aceitam movimentos para soluções de melhor qualidade numa vizinhança. As meta-heurísticas são vistas como uma evolução pois permitem uma piora na solução para fugir de locais ótimos.

De acordo com Osman (2002), uma meta-heurística é formalmente definida como um processo de geração iterativo que guia uma heurística subordinada, combinando os conceitos de diversificação e intensificação do espaço de busca, para encontrar boas soluções. O termo “diversificação” refere-se à exploração do espaço de busca, enquanto o termo “intensificação” refere-se à utilização da experiência acumulada durante este processo.

As meta-heurísticas têm sido aplicadas com sucesso aos mais variados problemas de otimização combinatória e inclusivamente a problemas reais, com provas dadas designadamente na resolução de instâncias de grandes dimensões (PINTO, 2004).

Existem diferentes maneiras de se classificar as metas-heurísticas, cada uma de acordo com um ponto de vista. Uma dessas importantes classificações considera o número de soluções utilizadas ao mesmo tempo em cada iteração, dividindo as meta-heurísticas em dois grupos: baseadas em trajetórias e baseadas em populações (BLUM et al., 2003).

Exemplos de métodos que se enquadram na categoria metas-heurística de trajetória são: Busca Tabu, *Simulated Annealing*, *Greedy Randomized Adaptive Search Procedure* GRASP, *Iterated Local Search* (ILS) e *Variable Neighborhood Search* (VNS). As metas-heurísticas Algoritmos Genéticos e Colônia de Formigas, são exemplos da categoria busca populacional.

2.6 Exemplos de Meta-heurísticas

As meta-heurísticas mais empregadas nos problemas de corte de estoque são: Busca Tabu, GRASP e Algoritmo Genético. Uma visão geral do estado da arte em meta-heurísticas pode ser encontrada em Blum et al. (2003).

A meta-heurística **Busca Tabu** (*Tabu Search*) é um procedimento utilizado para guiar um algoritmo heurístico de busca local (melhoramento) para explorar o espaço de soluções além dos ótimos locais. Seu principal componente é o uso da memória adaptativa para criar uma busca mais flexível. De acordo com Diaz (1996) na meta-heurística Busca Tabu, a idéia é utilizar estruturas de memória para guiar a busca por soluções vizinhas, guardando informações sobre a vizinhança percorrida nas últimas iterações. Este tipo de memória, chama-se “memória de curto prazo”, e serve para proibir que alguns movimentos sejam executados por um determinado período, para que a busca não retorne a uma solução já visitada. Existem ainda estruturas de memória chamadas de “memória de longo prazo”, que servem, por exemplo, para guardar soluções de elite (as melhores soluções encontradas), a fim de caracterizar atributos comuns às soluções de elite e que possam ajudar na exploração da vizinhança, e nos métodos de intensificação e diversificação.

A meta-heurística **Simulated Annealing** que também é conhecida como Recozimento Simulado. De acordo com Diaz (1996), seu procedimento de otimização inspirado no processo físico de resfriamento lento da matéria, ou seja, o algoritmo *Simulated Annealing*, na sua versão original, é utilizado em termodinâmica, para simular o processo de aquecimento de um sólido, seguido de um resfriamento progressivo e gradual, até atingir a temperatura ambiente, com o objetivo de alcançar o seu estado desejado, com energia mínima. O processo *Simulating Annealing* estabelece uma relação entre este tipo de comportamento termodinâmico buscando um mínimo global num problema de otimização. Neste caso, são permitidos movimentos que aumentem o valor da função objetivo, mas sua frequência é controlada por uma função de probabilidade que altera-se no decorrer da heurística.

Por outro lado, a meta-heurística **Algoritmo Genético** faz uma analogia à teoria de evolução de Darwin que incorpora uma solução potencial para um problema específico numa estrutura semelhante de um cromossomo e aplicam operadores de seleção, de *cross-over* e mutação, a essas estruturas de forma a preservar informações críticas relativas à solução do problema. Noronha et al. (2005) cita que uma implementação de um Algoritmo Genético começa com uma população aleatória de cromossomos, onde esta população são as soluções que vão

sendo encontradas no decorrer das iterações. Essas estruturas são, então, avaliadas e associadas a uma probabilidade de reprodução de tal forma que as maiores probabilidades são associadas aos cromossomos que representam uma melhor solução para o problema de otimização em relação àqueles que representam uma solução pior. A aptidão da solução é tipicamente definida com relação à população corrente.

2.7 GRASP

A meta-heurística (*Greedy Randomized Adaptive Search Procedure*) GRASP, é caracterizada por apresentar um método guloso (*greedy*) aleatório (*randomized*) e pela capacidade de adaptação da função gulosa (*adaptive*). Essa meta-heurística foi desenvolvida originalmente por Feo e Resende (1989), empregada para obter soluções em problemas de otimização combinatória.

Resende e Ribeiro (2003) conceituam a meta-heurística GRASP, como um algoritmo construtivo combinado com um método de melhoramento, baseado num processo iterativo, onde sucessivas soluções são construídas e melhoradas de uma forma independente.

Tipicamente consiste em realizar duas etapas: na primeira cria uma solução inicial adicionando elementos à solução, os elementos são escolhidos a partir de uma Lista Restrita de Candidatos (LRC) e na segunda etapa efetua uma busca local com o objetivo de melhorar a qualidade da solução. A melhor solução encontrada ao longo de todas as iterações é retornada como resultado.

Em Temponi et al. (2007), a meta-heurística GRASP consiste em usar diferentes soluções iniciais como pontos de partida para a busca local. Desta forma, o algoritmo executa, a cada iteração, a heurística de construção parcialmente gulosa, gerando uma solução, e, após, executa uma busca local nessa solução, gerando uma solução ótima local. Em seguida, é verificado se o valor, em termos de função de avaliação, dessa solução ótima local é melhor que o melhor valor encontrado. Em caso afirmativo, a solução corrente passa a ser a melhor solução.

A Figura 10 apresenta o pseudocódigo genérico da GRASP. Inicialmente recebe os dados que serão processados (linha 1) e repetidamente constrói uma solução (passo 3) e esta é melhorada por um procedimento de busca local (passo 4). A solução é atualizada se outra melhor for encontrada (passos 5).

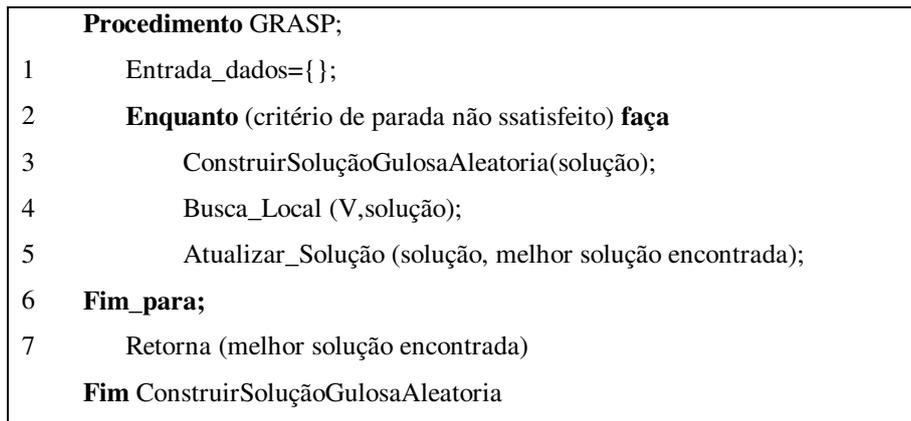


Figura 10 – Pseudocódigo genérico da meta-heurística GRASP.

A cada iteração, a GRASP apresenta uma solução e, para que isso não ocorra infinitamente, é necessário utilizar um critério de parada como:

- tempo de processamento.
- condição de otimalidade.
- número fixo de iterações;
- parar após o transcurso de um determinado tempo computacional;
- parar quando a melhora no valor da função objetivo não estiver sendo muito significativa.

O critério de parada na condição de otimalidade para um problema combinatório pode não ser trivial, pois esta solução na maioria dos casos não se conhece. Sendo os demais critérios mais usuais na literatura.

A idéia geral da meta-heurística GRASP, parte do seguinte princípio, quanto mais soluções viáveis diferentes são conhecidas para uma mesma instância, maior será a probabilidade de se achar a solução ótima ou então pelo menos se aproximar mais desta (Figura 11).

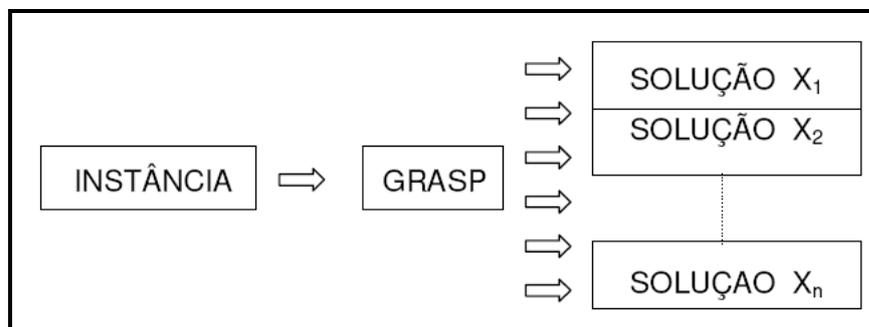


Figura 11 – A meta-heurística GRASP na solução do problema.

2.7.1 Fases da GRASP

A meta-heurística GRASP é composta por duas fases: a Construtiva e de Melhoramento.

a) Fase de construção

A fase de construção da GRASP utiliza uma heurística construtiva parcialmente gulosa. De acordo com Vieira Neto (2004), nas heurísticas gulosas o melhor elemento é aquele que minimiza ou maximiza a função de avaliação e é escolhido para compor a solução. Na heurística parcialmente gulosa, a escolha do próximo elemento é feita de forma aleatória dentre aqueles que compõem a LRC. As heurísticas parcialmente gulosas são caracterizadas pela escolha aleatória de um dos candidatos da LRC, enquanto as heurísticas gulosas são caracterizadas pela escolha do melhor elemento. Em Prais, (1999) construções parcialmente gulosas conduzem a soluções perto do ótimo local, acelerando a busca local.

A cada iteração, da fase construtiva, deve-se:

- usar um critério guloso e avaliar o benefício de incluir cada elemento na solução;
- criar uma lista restrita de candidatos, formada por elementos que são avaliados por uma função gulosa, escolhendo os melhores;
- aleatorizar e selecionar um elemento da lista de candidatos;
- adaptar a função gulosa com base nas informações geradas pelo elemento incluído;

A escolha do elemento a ser adicionado na próxima iteração do método construtivo, é feita colocando-se os elementos ainda não escolhidos em uma lista ordenada de acordo com a função gulosa.

As Restrições dos elementos na lista candidato podem ser baseadas em:

- número máximo de elementos na lista;
- qualidade dos elementos com relação a escolha gulosa.

A Figura 12 apresenta o pseudocódigo da fase de construção GRASP.

	Procedimento ConstruirSoluçãoGulosaAleatoria;
1	solução={};
2	Para solução_incompleta faça
3	CriarLRC;
4	Selecionar aleatoriamente um elemento da LRC(s);
5	solução=solução U {s};
6	Função_adaptaçã_gulosa;
7	Fim_para;
	Fim ConstruirSoluçãoGulosaAleatoria

Figura 12 – Fase de construção da GRASP.

b) Fase de melhoramento

A solução gerada pelo procedimento construtivo não garante otimalidade, é quase sempre vantajoso acrescentar um algoritmo de busca local, na tentativa de melhorar cada solução construída.

Neste procedimento, um método de busca local é aplicado, compara-se a solução da iteração anterior com a da iteração atual, descartando-se a pior, registrando-se a outra solução,

De acordo com Noronha (2005) na fase de busca local deve-se tomar cuidado em:

- escolher uma vizinhança apropriada;
- usar estruturas de dados eficientes para acelerar a busca local;
- ter uma boa solução inicial.

Na tentativa de melhorar as soluções originárias na fase de construção, emprega-se o procedimento de busca local, que verifica o espaço de soluções mudando de uma solução para uma solução vizinha. A Figura 13 apresenta o pseudocódigo da fase busca local, onde uma determinada estrutura de vizinhança V para o problema x , gerar a melhor solução $V(x_i)$, a partir da solução.

Procedimento Busca_Local (solução);	
1	Para solução faça
2	Encontrar melhor solução $x \in V(\text{solução})$;
3	Solução= x
4	Fim_para ;
5	Retorna (solução ótima);
Fim Busca_Local	

Figura 13 – Pseudocódigo Busca_local.

A partir de uma boa solução, produzida por um algoritmo guloso (aleatório), conduz-se para rápidas buscas locais. Os principais fatores que influenciam o desempenho e, que podem ser considerados parâmetros para um procedimento GRASP são:

- i) a função gulosa, a qual define a qualidade dos elementos incluídos na lista restrita, influenciando assim no desempenho do procedimento de busca local;
- ii) a estrutura de vizinhança e o procedimento de busca local, que definem o espaço de soluções (em função da solução inicial) a ser explorado e a maneira como esta exploração é feita;
- ii) a cardinalidade da lista restrita de candidatos, os quais definem juntamente com a função gulosa, o modo como as soluções iniciais são construídas e a aleatoriedade do método

A técnica GRASP pode ser implementada empregando filtros. De acordo com Resende (1998) o uso de filtros acelera a fase busca local. Em Temponi et al. (2007), demonstrou o uso de filtros, selecionando para a fase de busca local, somente a melhor soluções construídas de uma sucessão de iterações na fase de construção, ou seja, aplicar a busca local somente para soluções construídas pela fase de construção que atende algum critério de aceitação.

A técnica GRASP por sua característica de construção de soluções independentes, apresenta relativo sucesso para o tratamento de problemas de corte de estoque, destacam-se entre as metodologias utilizadas os trabalhos de Alvarez-Valdes et al.(2005) e Vieira Neto (2004).

Em Barros et al. (2004) uma das maneiras que se pode solucionar o PCEB de forma mais favorável é utilizar recursos que as heurísticas construtivas e gulosas nos oferecem e sucessivamente utilizar uma meta-heurística a qual venha orientar essas heurísticas e assim alcançar resultados ainda melhores.

A técnica GRASP é a base para solução deste trabalho, pois suas características atendem as perspectivas de solução para problema estudado, geralmente esta técnica produz boas soluções, apresenta poucos parâmetros a serem ajustados e rápido tempo de processamento.

2.8 Trabalhos Relacionados

As primeiras soluções para os problemas de corte de estoque tanto unidimensional quanto bidimensional foram baseada no método simplex com geração de colunas também conhecido como padrão *n*-grupo, este método foi proposto inicialmente por Gilmore e Gomory (1965). De acordo com Gomes et al. (2005) as dificuldades na resolução dos problemas de corte de estoque de médio e grande porte foram minimizadas graças aos trabalhos de Gilmore e Gomory (1965).

O método *n*-grupo ou *n*-estágio estudado por Gilmore e Gomory (1965), representa um arranjo de corte guilhotinado classificado como 2-estágios, este método divide as faixas resultantes do primeiro estágio em grupos de faixas, possibilitando que os grupos de faixas serem cortados simultaneamente no segundo estágio. Quando o arranjo apresentar apenas um grupo de faixas considera-se um arranjo tabuleiro ou 1-grupo. A característica principal deste método é que todos os itens apresentam-se distribuídos em linhas e colunas, num formato que lembra um tabuleiro de xadrez ou dama (Figura 14). Os padrões tabuleiros necessitam de pouco manuseio do objeto e, portanto, possuem um baixo custo operacional, porém podem apresentar altos índices de perda. Tal como os demais, um arranjo de corte *n*-estágio pode ser exato ou não-exato.

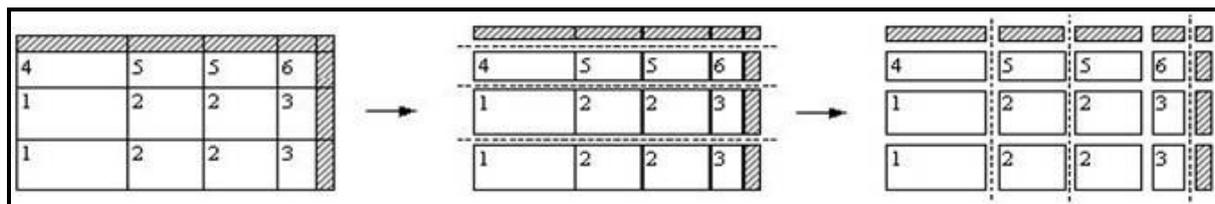


Figura 14 – Formato Tabuleiro. Fonte: Rangel e Figueiredo 2008.

Morabito (1994) examina alguns modelos de 2-estágios, e reporta uma experiência real da aplicação deste modelo numa indústria de papel e papelão (problema unidimensional) e numa indústria de móveis (problema bidimensional), modificando o modelo original e incluindo restrições práticas dos processos de corte, tais como, unidades de estoque com tamanhos diferentes e tolerância na demanda dos itens.

Rangel e Figueiredo (2008) analisaram os padrões de corte adotados por uma fábrica do pólo Moveleiro de Votuporanga-SP, com o objetivo de gerar padrões tabuleiros proposto por Gilmore e Gomory (1965). Desenvolveram uma heurística que analisa os padrões de corte utilizados pela fábrica e os converte para o formato de tabuleiro, removendo as faixas que impedem o corte em 2-estágio e considerando estas faixas novos objetos. Neste caso, qualquer parte da matéria-prima não aproveitada foi considerada perda.

Reeves (1996) trata a solução dos problemas empacotamento com o emprego da meta-heurística Algoritmo Genético, neste trabalho são descritos esquemas de mapeamento de cromossomos próprios para a resolução destes problemas. Os algoritmos híbridos são utilizados através do método algoritmos genéticos com os algoritmos de encaixe *Next Fit*, *First Fit* e *Best Fit*. Os resultados apresentados foram considerados satisfatórios apenas para pequenas instâncias.

Hopper (2003) reuniu os vários exemplos de problemas a partir de diferentes pesquisas e produziu resultados para todos os dados recolhidos. A técnica GRASP híbrida com *Bottom-left* é aplicada para problemas envolvendo formas regulares e irregulares.

No trabalho de Oliveira e Gomes (2002) uma heurística de dupla troca foi implementada e classificada como heurística de Busca Local. A partir de uma solução inicial obtida do algoritmo construtivo *Bottom-left*, trocas entre a vizinhança são realizadas. A escolha das peças a serem posicionadas é feita a partir de alguns critérios de ordenação, tais como, maior área, maior largura, maior altura, maior irregularidade ou aleatório. Após o posicionamento das peças foi aplicada a heurística de dupla troca (busca local) visando uma melhora na solução. Esta heurística simula a troca de peça com os vizinhos mais próximos. A partir de uma função de avaliação é definido o local onde o item será posicionado. Este método preenche “buracos” e melhora gradualmente a solução.

Barros et al. (2004) analisaram o problema de corte de estoque bidimensional buscando resolução através da meta-heurística GRASP. No procedimento para obter os padrões de corte é aplicado um algoritmo construtivo “guloso”. A partir dos padrões de corte um procedimento de busca local foi empregado, este procedimento remove as últimas peças que compõem o arranjo, posterior a esta etapa, o algoritmo busca novas peças visando otimizar o objeto. Observa-se que esta técnica apresenta solução com tempo computacional razoável.

Viera Neto (2004) apresentou a técnica GRASP aplicada a problemas de corte unidimensionais, propondo variações na fase de melhoramento. As soluções iniciais são geradas com emprego da GRASP original. Na fase do melhoramento os padrões de corte que apresentarem maior perda e maior variedade de itens são escolhidos. Nesta fase os itens são retirados da solução inicial e transformados em sub-problema. Os itens que compõem o sub-problema são submetidos a um gerador de padrões viáveis. Após a geração dos padrões viáveis, finaliza-se a solução aplicando o método construtivo.

Alvarez-Valdes et al. (2005) trabalharam com a técnica GRASP, onde a fase construtiva foi implementada de acordo com a técnica original, sendo que a cada iteração foi inserida uma peça afim de gerar os padrões de corte. Para melhorar estes padrões de corte foram apresentados três métodos: i) No primeiro método ocorre remoção do arranjo de corte blocos que obtiveram maior percentual de perda, os blocos restantes do arranjo de corte são reordenados. Para aproveitar o espaço livre do objeto, são selecionadas novas peças; ii) No segundo método, os blocos que apresentam maior perda são removidos do arranjo de corte e novas peças são inseridas, neste caso, os blocos restantes não sofrem ajustes (Figura 15); iii) No terceiro método são eliminados os blocos finais que compõem a solução, controlada por uma variável $K\%$ conforme o tamanho da instância uma quantidade blocos foi eliminada, um algoritmo construtivos foi aplicado para preencher os espaços vazios.

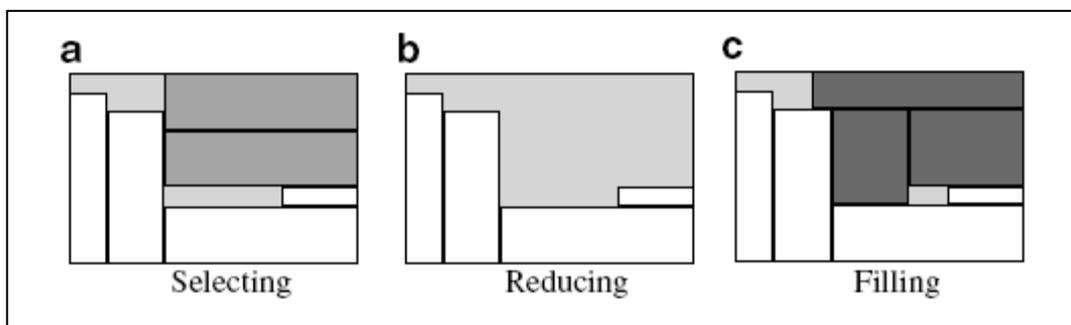


Figura 15 – Segundo método da busca local de Alvarez-Valdes et al (2005).

Velasco et al. (2009) estudaram o PCEB utilizando strings para definir os padrões de corte, por exemplo: F_H – designa uma faixa na horizontal, F_VPR – representa uma faixa na vertical contendo peças rotacionadas. A implementação da solução foi fundamentada na técnica GRASP

atuando diretamente no processo de formação e organização de faixas-guilhotinas. Na fase inicial utiliza-se um critério “guloso” selecionando as maiores peças para compor o arranjo de corte, a primeira peça inserida estabelece duas faixas guilhotinas a serem geradas por um corte guilhotinado horizontal e outro vertical (Figura 16). Posteriormente foi simulada a quantidade de peças a ser inserida nas duas faixas e a perda interna produzida. Na fase de melhoramento, analisaram os espaços ociosos (perdas) existentes a fim de preenchê-los. No final uma função de avaliação foi aplicada e a faixa que apresentar menor percentual de perda será incluída no arranjo de corte. Em testes, com seis instâncias proposta por Oliveira e Ferreira (1990) o algoritmo proposto obteve bons resultados.

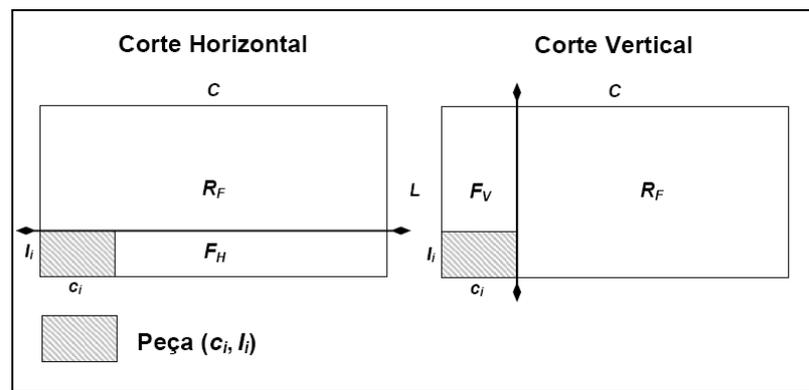


Figura 16 – Corte horizontal e vertical simulando a faixa.

Yanasse e Cerqueira (2009) propuseram uma variação da meta-heurística GRASP composta de três fases objetivando gerar padrões de corte menos distintos possíveis. Na primeira fase geram-se padrões sucessivamente que serão aceitos caso tenham desperdício limitado. Cada arranjo aceito é repetido o máximo possível, sem que itens sejam cortados além da demanda, priorizam-se itens grandes e com as maiores demandas. Na segunda fase, a solução foi melhorada, na terceira fase, uma técnica de redução de padrões é utilizada visando obter padrões de corte menos distintos possíveis.

Em Burke et al. (2007) empregaram as heurística foram Busca Local e a Busca Tabu para desenvolver uma aplicação para tratar do PCEG. As etapas iniciais do algoritmo são executadas de acordo com a técnica *Bottom-left*, posteriormente aplica-se busca local e os dados resultantes tratados com a Busca Tabu. Neste caso, o algoritmo encontra a solução rapidamente e com bom nível de precisão, tornando-o um forte candidato para aplicações industriais. O trabalho apresenta pontos iniciais de referência a problemas envolvendo os formatos circunferências e arcos de circunferência.

Oliveira e Ferreira (1990) tratam o PCEG através da GRASP pura onde a primeira fase da heurística consiste em gerar uma solução inicial através de um método construtivo guloso aleatorizado, na segunda etapa uma heurística de busca local foi aplicada sob a solução gerada pelo método construtivo, a cada iteração se uma solução melhor que a solução corrente for encontrada então a solução corrente foi atualizada.

2.9 Modelagem Geométrica

As soluções para o Problema de Corte de Estoque Bidimensional requerem o uso de técnicas de modelagem geométrica. Em geral, as linguagens de programação empregada são C e C++ (ARAÚJO 2006, HOPPER & TURTON 2001). Estas linguagens oferecem suporte para desenvolver aplicações com tempo computacional competitivo, porém não apresentam recursos para tratar da componente geométrica que envolve este problema, obrigando a implementação de cálculos complexos ou o uso de bibliotecas geométricas. As bibliotecas geométricas como, por exemplo, LEDA - *Library of Efficient Data types and Algorithms* são desenvolvidas em códigos abertos e de fácil acesso, mas geralmente precisam ser ajustada para integrarem modelagens específicas.

Segundo Costa (2007) a complexidade dos PCEB é refletida na necessidade de garantir uma precisão adequada na verificação da existência de sobreposição no posicionamento de dois itens e devem ser implementadas de uma forma eficiente, uma vez que são recursos elementares na resolução deste tipo de problema. Assim, o emprego de técnicas de interação gráfica capaz de detectar quando duas peças ocupam o mesmo espaço é fundamental para gerar padrões de cortes factíveis.

De acordo com Gomes (2008), a dificuldade de implementar funções eficientes para lidar com a componente geométrica do problema provoca a falta de precisão e simplificações excessivas nas formas geométricas. Também é habitual restringir as orientações admissíveis para cada peça a um conjunto discreto e pequeno, devido a complexidade de implementar rotinas que lidem eficientemente com rotações contínuas. Neste contexto, as ferramentas de Desenho Auxiliado por Computador (CAD) surgem como uma alternativa para suprir as dificuldades relacionadas à modelagem geométrica que decorre em implementações de soluções para o Problema de Corte de Estoque Bidimensional

Os sistemas CAD, foram desenvolvidos para substituir o desenho manual e até meados de 1950 sua aplicabilidade se limitava à elaboração de desenhos técnicos (FILHO, 1999). De acordo com Souza e Coelho (2003) nesta época os sistemas CAD já propiciavam várias vantagens como: possibilidade de envio, ou recebimento de desenhos por processos eletrônicos; melhor gerenciamento dos desenhos; precisão do dimensionamento; maior rapidez na recuperação, modificação ou atualização de desenhos.

A utilização dos sistemas CAD até a década de 90 era limitada a aplicação em grandes empresas, como aeroespacial e automobilística, devido aos custos de software e hardware que a utilização destes recursos demandava. A popularização do computador e a evolução dos equipamentos informatizados propiciou preços bastante acessíveis aos softwares CAD tornando sua utilização viável mesmo para pequenas empresas (KONG, 2003).

Atualmente as ferramentas CAD são consideradas de forma bastante ampla, uma tecnologia multidisciplinar, ou conjuntos de ferramentas utilizadas por todas as áreas em que existe uma forma desenvolvida de interação do computador à atividade de projeto, bem como ao controle e gestão deste processo, permitindo que as informações geométricas que constituem o desenho sejam criadas, manipuladas e apresentadas graficamente na tela do monitor e, ao mesmo tempo, armazenadas numericamente na memória do computador.

De acordo com Simões (1999) a modelagem CAD pode assumir uma função mais nobre e inteligente num ambiente de manufatura, quando sua função não está vinculada simplesmente a representação gráfica de um objeto, mas assume outras aplicações como, análise estrutural, verificação de interferência entre conjuntos. Além disso, o mesmo modelo CAD pode servir de base para outras aplicações.

Segundo Nakamura et al. (2003) a utilização de sistemas computacionais nas diferentes áreas da engenharia se tornou uma exigência do atual mercado globalizado altamente competitivo, onde as empresas necessitam produzir cada vez mais, melhor e com custos reduzidos. Além de simplesmente substituir a prancheta pelo computador – automatizando o trabalho do desenhista – os atuais sistemas CAD são capazes de aumentar a eficiência de praticamente todas as atividades realizadas por um departamento de engenharia; onde as diferentes etapas do processo de engenharia desenvolvem-se de maneira integrada e simultânea.

Os pacotes CAD apresentam um potencial que pode ser considerado ilimitado por oferecem comandos como: produzir automaticamente uma visão de *design*; selecionar entidades no desenho e obter o ponto de rotação de uma posição para outra; mover, copiar, excluir e alongar facilmente entidades dentro do desenho (MARTÍNEZ, 2004).

O sistema CAD, bem implantado, pode aumentar significativamente a produtividade do departamento de projetos, através da implantação de vários tipos de técnicas complementares: personalização do CAD, transformando rotinas do dia a dia de trabalho em formas práticas de utilização.

No ambiente CAD, as informações geométricas são representada analiticamente através de pontos, linhas, cones, planos, superfícies quadráticas, etc., e por funções booleanas. Essas representações são geradas interativamente quando o usuário modela a geometria da peça na interface do CAD. Sendo fundamental que os softwares de CAD ofereçam ferramentas para a modelagem de geometrias complexas (SCHIMIT, 2002).

A tecnologia computacional aplicada a sistemas CAD proporciona a manipulação de entidades gráficas de forma a verificar interferências entre peças, problemas de montagem, espaço de trabalho, dimensões, tolerâncias, enfim, uma automação capaz de minimizar os erros de projeto e proporcionar a visualização de uma simulação do produto. A modelagem de um projeto pode assumir diferentes formas de representação nos sistemas CAD, entendida como modelagem geométrica. As técnicas de modelagem são classificadas em três grupos: Modelagem de Sólidos, Modelagem de Superfície e Modelagem Híbrida (MONARO e SCHUTZER, 2005).

2.9.1 Modelagem de Sólidos

A modelagem geométrica por sólidos concentra-se na representação dos aspectos estruturais do produto. Esta modelagem é realizada a partir de formas simples bidimensionais como polígonos, círculos ou a combinação destes. Dependendo da geometria desejada, o sólido pode ser gerado fazendo-se a forma bidimensional percorrer uma trajetória ou ainda pela conexão de duas ou mais destas formas, desenhadas em planos diferentes.

Zeid (2005) define um modelo sólido como uma representação completa, única e livre de ambigüidades. Existem vários métodos de modelagem sólida, sendo que os principais são: *Features*, Paramétrica, *WireFrame*, CSG (*Constructive Solid Geometry*) e B-Rep (*Boundary Representation*).

As *features* (características) podem ser definidas como elementos físicos das formas geométricas que tem algum significado para a engenharia, e a modelagem sólida paramétrica permite a geração de modelos com dimensões vinculadas à variáveis, permitindo a regeneração automática do modelo após cada modificação. Também possibilita todas as dimensões de um produto modelado por um sistema CAD estejam relacionadas entre si, através de um parâmetro. Quando se altera o valor numérico deste parâmetro, todos os valores atrelados a ele se alteram

automaticamente. Os sistemas com esta característica podem ser aplicados na elaboração de uma família de peças (SPECK, 2005)

O método *wireframe* se ocupa da representação de objetos geométrico para sistemas 2D a partir de entidades primitivas que são formas geométricas básicas (pontos, retas, centros e vértices de entidades planas, combinação dessas entidades) onde a representação gráfica é feita por esquemas, possibilitando a geração de modelos baseados em símbolos, basicamente empregado para modelar objetos nos quais só interessa o contorno, perfis, trajetórias, sapatos, carrocerias, fuselagens (ZEID, 2005). Este método é útil para manipulação dinâmica, como por exemplo, a animação dos movimentos de mecanismos, devido à sua estrutura de dados simplificada, que também favorece a realização rápida de operações de manipulação, edição, inserção, modificação, e deleção de elementos do modelo (CUNHA, 2000).

Os *Constructive Solid Geometry* (CSG) são métodos capazes de gerar objetos dimensionais e tridimensionais, possuindo centro de gravidade e volume, utilizados para a modelagem de sólidos mais complexos, possibilitando a utilização de comandos baseados em operações booleanas como união, subtração e interseção. Também oferece suporte a transformações geométricas (rotação, translação, escalamento) tornando a modelagem de produtos com este método bastante ágil (CUNHA, 2000). A modelagem sólida não armazena só a geometria do objeto final, mas também todas as formas primitivas e operações usadas para a sua construção (FOGGIATTO et al., 2007).

A representação *Boundary Representation* (*B-Rep*) armazena detalhes de como as faces, arestas e vértices de um sólido que se unem para representar um objeto. Um modelo *B-Rep* que representa um sólido possui a capacidade de descrever como cada face está conectada as suas faces adjacentes, de modo a formar um volume totalmente fechado no espaço. As informações geométricas relacionada aos objetos sólidos formam os constituintes básicos dos modelos *B-Rep* (TAKASE et al., 2005). A estrutura de dados deste modelo segue uma organização hierárquica, formada por faces, arestas, ciclos, e vértices (Figura 17).

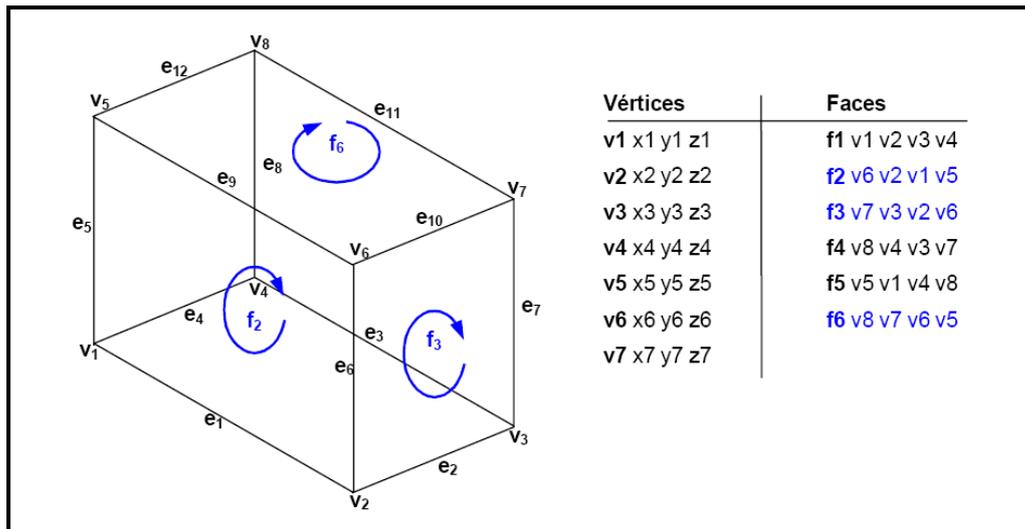


Figura 17 – Estrutura de Dados para o modelo B-rep. Fonte Cunha 2000.

2.9.2 Modelagem de Superfície

Os Sistemas CAD que utilizam modeladores de superfícies empregam formulações matemáticas complexas, conhecidas como funções *Spline*. Segundo Souza (2003), neste tipo de modelagem as geometrias não possuem espessura. As posições (direções X, Y, Z) de cada ponto, podem ser alteradas, mantendo o objeto tridimensional (Figura 18). Isso possibilita o modelamento de formas complexas. A modelagem por superfícies é indicada quando o modelo possui curvas complexas ou superfícies livres (FOGGIATTO et al., 2007). Em Cordeiro e Másculo (2005), uma superfície pode ser definida como um elemento matemático que separa o interior do exterior de um objeto. Os objetos tridimensionais gerados pela técnica de modelos de superfície diferem dos modelos por *wireframe*, por usarem superfícies tridimensionais, definindo um volume ou contorno de um objeto.

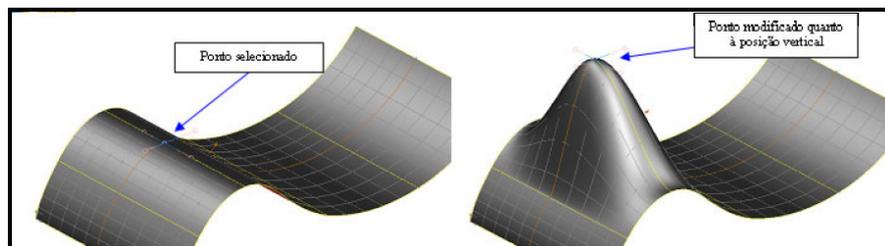


Figura 18 – Exemplo de modelamento em superfície. Fonte Souza 2003.

Um sistema CAD modelador de superfícies não permite a utilização de operações booleanas para a intersecção entre geometrias, fato que torna o trabalho mais penoso. Em geral,

os softwares desta classe possuem custo mais elevado, devido à complexidade destes sistemas, se comparado aos modeladores de objetos sólidos. Contudo, estes softwares permitem a geração formas complexas, que são requeridas em determinadas aplicações (SOUZA et al., 2003).

2.9.3 Modelagem Híbrida

Os sistemas CAD que utilizam modeladores híbridos podem ser considerados sistemas mais robustos que utilizam complexos algoritmos matemáticos. Estes sistemas permitem a utilização de qualquer um dos métodos citados acima, possibilitando atender todas as necessidade de modelar um determinado produto (CARDOSO, 2009).

Os sistemas CAD tipo geral permitem ainda acoplar módulos de simulação específicos para determinadas áreas. Como é o caso do AutoCAD, 3D-Studio e Microstation-Modeler, BRL-CAD, ACARD.

2.10 Sistemas CAD/CAM

Os Projeto e Manufatura Auxiliados por Computador (CAD/CAM) envolvem o uso de sistemas de computador para realizar certas funções em desenho e fabricação. Nestes sistemas, os ambientes CAD envolvem o uso de computadores para facilitar a função de desenho e as ferramentas CAM apóiam as atividades de manufatura. Esses dois foram unidos em uma definição comum denominada CAD/CAM porque o segundo ato é como uma atividade continuada do primeiro e ambos são integrados pelo desenho e manufatura.

Um sistema CAD/CAM integrado pode ser definido como o sistema em que a fabricação e o projeto foram ligados através de dados e informações acumuladas no processo de projeto e transmitidos diretamente para os procedimentos de manufatura (SIMÕES, 1999).

A seqüência convencional de desenvolvimento de produtos em sistemas CAD/CAM inicia-se com a modelagem geométrica do produto utilizando um *software* CAD. A informação gerada em CAD (modelo conceitual) pode posteriormente ser exportada e importada para um sistema CAM (permitindo, por exemplo, gerar as trajetórias de usinagem), devido estes sistemas permitem a associatividade de dados (base de dados única) onde a informação do projeto é partilhada entre as diversas aplicações, sem que cada aplicação tenha de traduzir ou transferir

dados. Esta integração entre aplicações evitando a perda de informações bastante comum neste tipo de transferência (NAKAMURA et al., 2003).

A fabricação de um móvel onde o projeto e o *design* do produto para a geração e detalhamento do molde são desenvolvidos em *software* CAD, e após a elaboração destes moldes um sistema CAM gera os programas de Comando Numérico (CN) que serão executados em uma máquina de Comando Numérico Computadorizado (CNC) produzindo os cortes do móvel é um exemplo prático de integração entre os sistemas CAD/CAD e como as formas geométricas construídas com *software* CAD estão ligadas em outros sistemas computacionais (Figura 19).

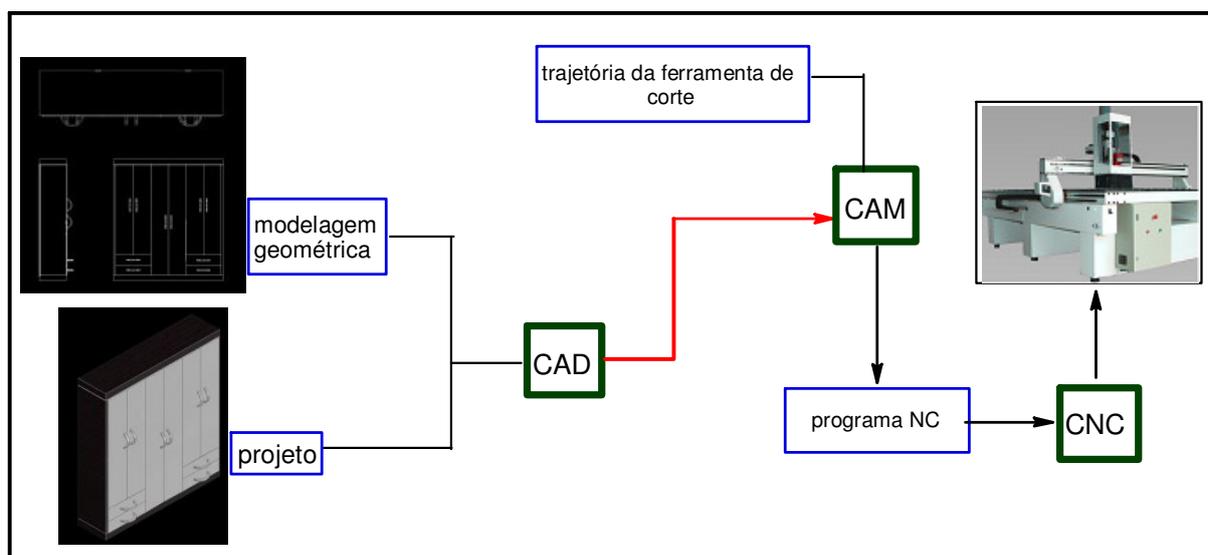


Figura 19 – Sistemas CAD/CAM operando de forma integrada.

Uma vez modelado o produto, pode-se escolher o processo de fabricação, selecionar a máquina, informar a seqüência de fabricação, escolher ferramentas e condições de corte. A partir disso o sistema calcula, automaticamente, a trajetória das ferramentas, transferindo-as, na forma de um programa CNC, para determinada máquina-ferramenta (COSTA e PEREIRA, 2006).

De acordo com Meneghello (2003), os recursos de auxílio à manufatura permitem que dados do produto, como geometria e dimensões, gerados a partir de sistemas CAD, possam ser usados diretamente na geração de programas de comando numérico para máquinas CNC, planejamento de processo e controle de produção.

Os benefícios dos atuais sistemas vão além da integração da manufatura como por exemplo: síntese de grandes volumes de dados; facilidade na geração e alteração de imagens; simplificação de diversas atividades de projetos; produção de desenhos de melhor qualidade e tempo reduzido; precisão de cálculos de projeto onde bibliotecas matemáticas provêm rapidez na resolução dos cálculos; base para a análise e fabricação dos produtos (CUNHA, 2000).

Ainda segundo Cunha (2000) a interpretação da geometria das peças através da leitura de arquivos gráficos de transferência de dados e a geração de programas CN ocorrem através da interatividade com o programador, pois estes sistemas possibilitam a seleção de parâmetros de operação, ferramentas e tipo de movimento em um banco de dados customizável. Com a interpretação dos dados geométricos e as informações do banco de dados resultante da interatividade, o sistema calcula e define as estratégias de usinagem como trajetórias e parâmetros de operação. Estas informações são então armazenadas em um arquivo de dados para posterior pós-processamento.

Uma das grandes vantagens das indústrias que fazem uso de sistemas CAD/CAM é ter seus processos de desenvolvimento e fabricação integrados, ou seja, após definir a modelagem do produto é possível enviar os resultados para equipamentos automatizados, como as guilhotinas CNC, fazendo conexão entre os desenhos no sistema CAD e a máquina de corte (TEBIS, 2006).

2.11 Comando Numérico Computadorizado (CNC)

O termo CNC significa máquinas controladas por computador, ou seja, são máquinas às quais se informa alguns parâmetros para sua operação, que passam então a ser executados; aduzindo-se o computador e nos últimos anos conseguiram-se avanços significativos quanto às possibilidades e facilidades de manipulação das máquinas industriais (NEVES, 2005).

O CN (Controle Numérico), basicamente compreende o comando dos movimentos de aproximação, avanço e recuo de uma máquina (COSTA e PEREIRA, 2006). O programa CNC transmite à máquina, por exemplo, a geometria de uma peça e as informações relativas à movimentação das ferramentas necessárias a execução do corte desta peça (SPECK, 2005).

A tecnologia CN, associada à modelagem digital encontrada em sistemas CAD/CAM, suporta em grande parte a transferência do modelo de um produto para a máquina com pouca intervenção humana, além de propiciar a substituição do meio de transmissão, papel ou verbal, para o eletrônico (COSTA e PEREIRA, 2006). Um CNC de arquitetura aberta que pode receber informações de planejamento e projeto em linguagem de alto nível e executar tarefas de acordo com a máquina a ser operada.

A utilização de máquinas CNC tem uma papel importante a flexibilização da produção, posterior teste e aprovação do programa possibilitam produzir um, ou mil produtos idênticos com grande precisão e repetibilidade. Conseqüentemente um produto pode ser substituído facilmente por um próximo tipo a ser produzido, havendo trocas rápidas na preparação da máquina (*setup*)

(NEVES, 2005). Nos processos de fabricação automatizados, as máquinas CNC têm sido amplamente empregadas, tanto para produção de peças complexas como para produção de pequenos lotes.

Em 1982, a Organização Internacional para Normalização (ISO) estabeleceu os princípios básicos da programação CNC (norma ISO 1056). A norma indica o formato básico do programa, de modo que um conjunto de comandos, compostos de palavras-chave, possa dar instruções para o sistema de controle.

As instruções podem referir-se à uma movimentação específica dos eixos da máquina, à uma indicação de sentido de giro do eixo (x, y e z) ou mesmo a um pedido de troca de ferramenta. A linguagem de programação é feita através de linhas de comandos (POLL, 2008). Um programa CNC é escrito em linguagem própria dos fabricantes de comando ou de forma padronizada. A Tabela 2 apresenta comandos padronizados pela norma ISO 1056.

Tabela 2 – Comandos CNC

Código	Significado
O	Número do programa
N	Número da linha do programa
G	Comandos de deslocamento e cálculo
M	Funções auxiliares
F	Velocidade de avanço
T	Chamada de ferramenta
S	Rotação de fuso
X/Z	Coordenadas em valores absolutos
I/J/K	Parâmetros de interpolação

Fonte: ISO 1056.

Dependendo do tipo de máquina-ferramenta e da quantidade de eixos comandados, o padrão G é suficiente para a fabricação de uma gama elevada de componentes mecânicos. Como exemplo, pode-se citar os tornos com apenas dois eixos (X e Z), os quais permitem a obtenção de várias formas geométricas definidas por superfícies de revolução. Neste sentido, o conjunto de *softwares* (CAD-CAM-CNC) é condição necessária, para que uma dada empresa possa ter um sistema integrado de manufatura (COSTA e PEREIRA 2006).

2.12 Aplicativos CAD

A finalidade de aplicativos CAD é a personalização do programa, para atender necessidades específicas, para usar melhorias desenvolvidas por terceiros e também para possibilitar o desenvolvimento próprio e a posterior comercialização.

De acordo com Souza 2006 as linguagens nativas dos sistemas CAD possibilitam que as funções sejam atualizadas e aprimoradas constantemente e raramente funções antigas tornam-se obsoletas. Na maioria das vezes, as funções antigas trabalham da mesma maneira na última versão e nas versões anteriores.

As funções implementadas em ambientes CAD possibilitam a automatização de uma sequência de comandos específicos de um determinado processo, essas funções também podem ser desenvolvidas em forma de aplicativos elaboradas no ambiente computacional de projeto(CAD). Segundo Da Silva e Bevilacqua (2005) para utilização dos sistemas CAD/CAM comerciais é necessário investir no desenvolvimento de rotinas para executar funções específicas de integração CAD/CAM. Implementar rotinas CAD permite adaptá-las às características específicas do equipamento onde o sistema será aplicado.(Da Silva e Souza, 2003; Gonçalves e Da Silva, 2008).

3. METODOLOGIA

A metodologia deste trabalho apresenta o emprego da técnica GRASP e descreve como empregar os recursos do CAD na implementação de soluções para o PCEB. Desse modo, a técnica GRASP foi empregada visando buscar a melhor combinação entre as peças de acordo com uma determinada demanda, enquanto, os recursos CAD foram utilizados na resolução das peculiaridades de modelagem geométrica do problema.

Porém, antes de detalhar a meta-heurística GRASP e as ferramentas CAD, é necessário apresentar o critério de ordenação das peças, a técnica de posicionamento e a modelagem computacional da solução.

3.1 Critérios para ordenação das peças

As heurísticas construtivas utilizam critérios de ordenação baseados nas características das peças. Estes critérios privilegiam certas características das peças tais como: área, altura e comprimento. Portanto, as peças são ordenadas de acordo com a características desejada.

Gomes (2005) classifica os critérios de ordenação como estáticos e aleatórios. Os critérios estáticos são baseados nas características geométricas dos itens e geram as seqüências ordenadas priorizando uma característica específica dos itens como:

- maior área;
- maior perímetro;
- maior comprimento;
- maior largura;
- maior índice agregado, neste caso, as peças recebem uma prioridade como, por exemplo, a demanda.

Neste trabalho, o desenvolvimento do algoritmo empregou-se o critério de ordenação estático de maior área. Este critério permite otimizar os espaços na chapa.

3.2 Técnica *Bottom-Left*

As técnicas de encaixe *Next-fit*, *First-fit*, *Best-fit*, *Last-fit*, *Worst-fit* e *Bottom-Left* são classificadas como heurísticas construtivas e permitem padronizar a forma com que os itens são posicionados na chapa. Devido sua característica peculiar do problema em questão, a técnica *Bottom-left* foi empregada neste trabalho.

A técnica *Bottom-left* trabalha posicionando as peças, inicialmente, no canto superior direito da chapa, sendo inseridos na seqüência em que aparecem na lista de peças. Em seguida, cada item é deslocado verticalmente para baixo até atingir o limite da chapa ou outro item. Após o item é deslocado horizontalmente para esquerda até atingir outro item ou o limite da chapa. Esse movimento para baixo e para a esquerda se repete até que não seja mais possível movimentar os itens (CONSTANTINO et al., 2002). A Figura 20 ilustra à heurística *Bottom-left*.

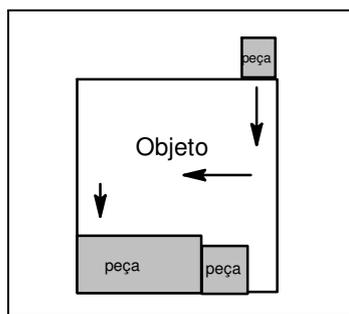


Figura 20 – Técnica *Bottom-left*.

3.3 Estruturas para a construção do algoritmo

As soluções para problemas de natureza combinatória, como é o caso do PCEB, empregando técnicas heurísticas, necessitam inicialmente, que seja definida uma modelagem computacional adequada visando facilitar a resolução desta classe de problemas. A modelagem proposta neste trabalho é composta por cinco partes:

- a) PEÇA: esta estrutura representa um item que faz parte da demanda para confecção de um determinado produto. Na implementação, esta estrutura é um vetor que armazena quatro variáveis i , c , l e a que representam, respectivamente, o índice, o comprimento, a largura e

a área. Sendo que a variável área é armazenada visando facilitar o acesso desta informação.

- b) CHAPA: a estrutura chapa representa a matéria-prima utilizada para produção da demanda. Este elemento é composto por três variáveis i , C , L que designam as informações de: índice, comprimento e largura do objeto, respectivamente.
- c) LISTA_PEÇAS: esta estrutura é composta por todos os elementos PEÇA com seus respectivos pontos de referência. O ponto de referência de uma peça é aquele ponto que estiver mais a esquerda e mais a baixo.
- d) LISTA RESTRITA DE CANDIDATOS (LRC): esta estrutura é composta por peças ordenadas, de acordo com o critério estático de maior área. A quantidade de peças na LRC varia conforme a dimensão do problema, obedecendo à seguinte regra:

$$\text{Nº de Elementos} \begin{cases} \text{Se total de elementos } (Te) \text{ for maior que } 100 \text{ então } Te / 4; \\ \text{Caso contrário } Te / 2. \end{cases}$$

Onde, Te é a quantidade de peças da que compõem a *Lista_peças*.

- e) SOLUÇÃO: a estrutura SOLUÇÃO de dados é um vetor que contém os elementos CHAPAS e os elementos PEÇAS necessário para formar os arranjos de corte (Figura 21).

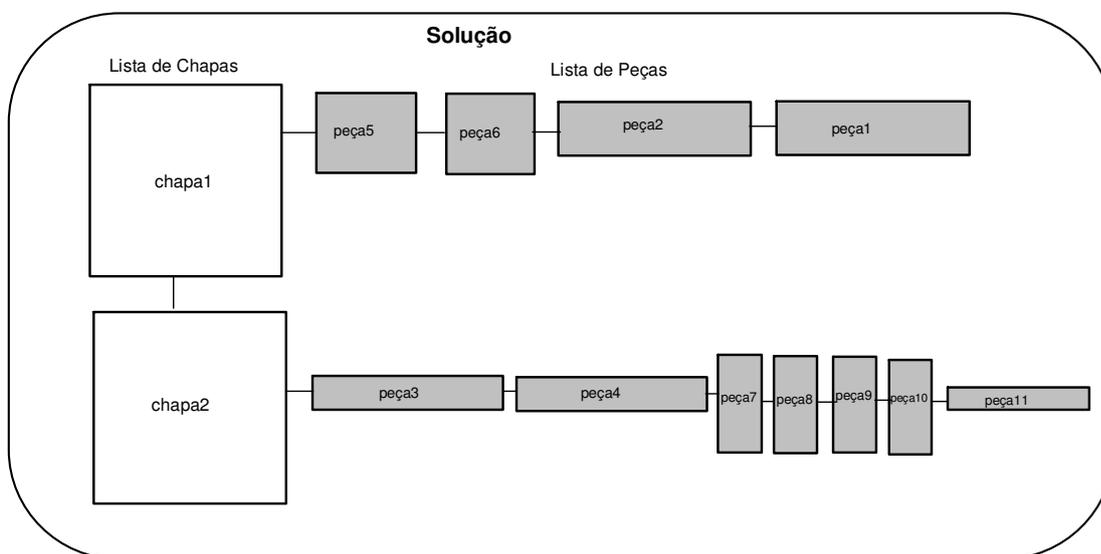


Figura 21 – Exemplo da estrutura SOLUÇÃO para o PCEG.

3.4 Aplicação da meta-heurística GRASP ao PCEB

A resolução dos problemas de corte de estoque dependem da otimização combinatória, visando encontrar a melhor combinação de acordo com o objetivo em estudo. Para este problema combinatório o uso de métodos exatos torna a resolução demorada e dependendo do número de combinações impossível de se realizar. Em decorrência desta situação, é proposta a aplicação da técnica GRASP para a solução do PCEB. A técnica GRASP atua diretamente no processo de arranjo das peças e na formação e organização das faixas guilhotinadas. Durante o processamento da meta-heurística é necessário analisar as soluções parciais e finais. O algoritmo proposto emprega nestas análises uma função de avaliação e um cálculo de desperdício.

A função de avaliação (f_{av}) é o somatório das variáveis áreas (a) das correspondentes peças de uma solução, também apresenta o número de chapas utilizadas. Como o objetivo do PCEB é utilizar o menor número possível de chapas para produzir a demanda de peças, então, quanto maior o valor da função de avaliação, melhor será a solução. O desperdício pode ser calculado através da subtração da área da chapa pelo somatório de todas as variáveis áreas que compõem uma solução. A fórmula 1 apresenta o cálculo do desperdício:

$$\text{Desperdício} = \{ \text{área.chapa} - \sum (a_1+a_2+a_3\dots a_n) \} \quad (1)$$

No desenvolvimento do algoritmo é utilizada uma função gulosa (f_{gul}) responsável por selecionar, aleatoriamente, da LRC as peça para formarem o arranjo de corte. A primeira peça selecionada define uma faixa no sentido horizontal. Os arranjos de corte desta implementação aceitam faixas somente no sentido horizontal. Portanto, o arranjo de corte de todas as chapas é composto por faixas de altura variáveis no sentido horizontal.

O critério de parada utilizado na implementação foi o número de iterações ($itermax$). O $itermax$ varia de acordo com a dimensão do problema. A formula 2 (TEMPONI, 2009) apresenta esta variação:

$$Itermax = \begin{cases} n/5, & \text{se } n/5 > 20 \\ 20, & \text{caso contrário} \end{cases} \quad (2)$$

sendo n o número de peças consideradas.

Como foi apresentado na Seção 2.7, a meta-heurística GRASP é composta por duas fases: a fase de construção e a fase de melhoramento.

3.4.1 Fase de Construção

O problema de corte de estoque pode ser exemplificado por uma chapa retangular estocada, de comprimento C e largura L ($Chapa = C, L$), em quantidade suficiente para atender a demanda, e um conjunto de peças retangulares de comprimento ci e largura li $\{Lista_peças (c1, l1), (c2, l2), \dots, (cn, ln)\}$, com demandas di a serem atendidas. Neste caso, a execução dos cortes guilhotinados ocorre paralelos ao eixo que representa a coordenada X.

Assim, uma peça (ci, li) alocada no canto esquerdo inferior da $Chapa = (C, L)$, determina uma faixa e um corte horizontal (FH) a ser executados. Na fase de construção da meta-heurística GRASP para o PCEB faixas guilhotinas horizontais são construídas e preenchidas com peças, definindo um arranjo de corte. (A Figura 22 demonstra o processo de geração de faixas). A cada iteração desta fase são executados os seguintes procedimentos:

Etapa 1 – Execução da função gulosa ($fgul$) para definir a LRC. A LRC será composta de peças que apresentam maior área, ordenadas de maneira crescente;

Etapa 2 – Seleção aleatória de uma peça (pk) da LRC;

Etapa 3- Inserção da pk na chapa definindo uma faixa horizontal (FH);

Etapa 4 – Adaptação da função gulosa com base no comprimento da faixa, configurando uma nova LRC, ou seja, a LRC, neste momento, é composta de peças que apresentem comprimento iguais e inferior ao comprimento da faixa;

Etapa 5 – Seleção de outra pk de acordo com a nova configuração. Caso não existam peças com comprimento igual, a $fgul$ aceita peças com dimensão imediatamente inferior ao da faixa. As peças também podem sofrer rotação de 90° ;

Etapa 6 – Inserção da peça na faixa, de acordo com a técnica *Bottom-left*;

Etapa 7 – Atualização do comprimento da faixa (FC);

Etapa 8 – Execução das etapas 5 a 7, enquanto a $Lista_peça$ permitir condições para o critério guloso, ou seja, enquanto a $Lista_peça$ for diferente de zero ou existir espaços na faixa;

Etapa 9 – Aplicação da função de avaliação (fav) na faixa e incremento da variável i que representa o número de iterações.

Enquanto, i for menor que $intermax$ os procedimentos de 1 a 9 são simulados, repetidamente. Ao final de cada iteração o valor da função de avaliação corrente é comparado com o valor anterior. Se o valor for considerado maior, o valor da fav é atualizado.

A faixa que obtiver melhor valor na função de avaliação é inserida na solução e o comprimento da chapa C é atualizado. Outras faixas serão geradas enquanto o comprimento da

chapa permitir ou enquanto a demanda for maior que zero. Quando a chapa estiver preenchida e a demanda for maior que zero, uma nova chapa será gerada. Esse processo iterativo repete-se até que a demanda corresponda à zero ($di = 0$).

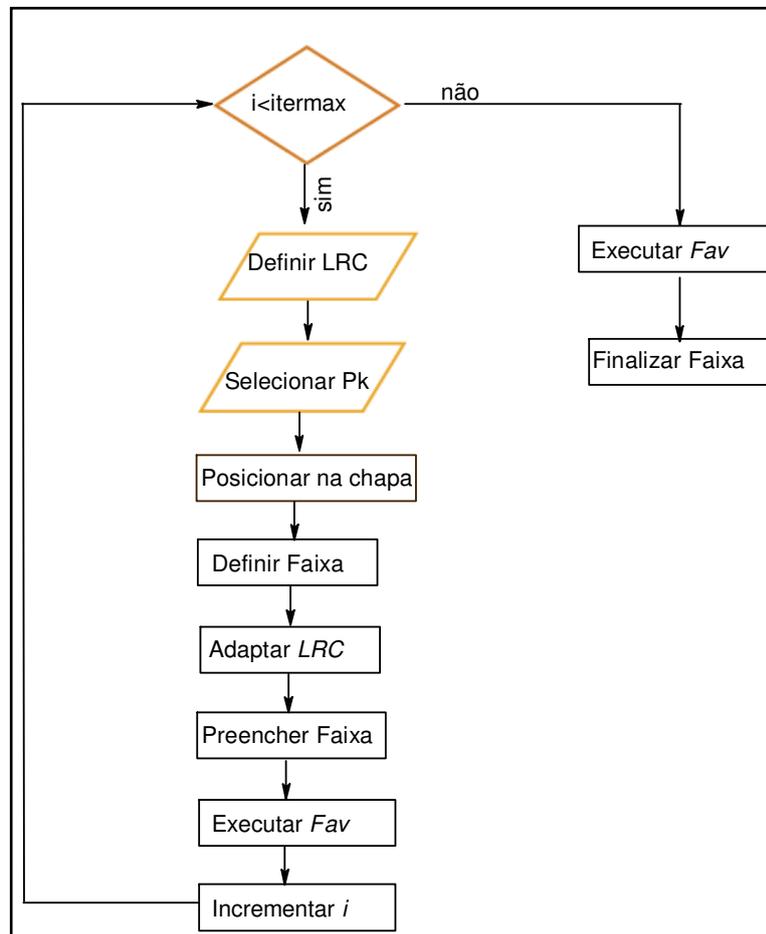


Figura 22 – Processo para geração de faixas horizontais.

3.4.2 Fase de Melhoramento

Tradicionalmente, a meta-heurística GRASP emprega, ao final das fases construtivas, a etapa de melhoramento. Porém, a meta-heurística permite o uso de filtros que possibilitam aplicar o algoritmo de melhoramento apenas na melhor solução construída da execução do algoritmo construtivo.

Tendo em vista que a faixa produzida na primeira fase da GRASP não fornece garantia de aproveitamento máximo do espaço na faixa, aplica-se a fase de melhoramento somente na melhor

faixa oriunda da fase construtiva. Assim o processamento da meta-heurística é acelerado diminuindo o tempo computacional necessário.

A fase de melhoramento aproveita os espaços da faixa horizontal corrente minimizando o valor percentual da perda gerada. A partir da faixa proveniente da fase de construção, a meta-heurística GRASP executa os seguintes passos:

- i. Identifica os espaços vazios na faixa;
- ii. Varre a *Lista_peças* na procura de peças candidatas a efetuar uma melhoria na respectiva faixa *FH*, que não devem possuir medidas (*li* e *ci*) que extrapolem as dimensões do espaço não preenchido;
- iii. Altera a orientação das peças em 90°, caso a *Lista_peças* não satisfizer a condição do passo ii;
- iv. Posiciona a peça selecionada no espaço não preenchido;
- v. A área disponível é atualizada, o passo ii é executado repetidamente até que houver espaço não preenchido ou enquanto existirem peças candidatas a serem testadas.

3.5 Representação Geométrica

Nas resoluções para PCEB, o principal objetivo é encontrar a melhor combinação das peças de um determinado objeto formando um arranjo que deve ser apresentado graficamente. Neste caso, os recursos utilizados na implementação da solução devem fornecer suporte para: usar variáveis, efetuar cálculos de dimensões e cálculos de massa, empregar operações booleanas, representar graficamente as formas geométricas, acessar as informações do desenho, e modificar os dados. Desse modo, a metodologia proposta, utiliza ferramentas de um sistema CAD que possibilita o desenho dos componentes do problema, bem como, acesso e manipulação das propriedades do desenho.

No processo de arranjo de corte é necessário representar geometricamente os componentes. Para isso, as ferramentas de desenho disponibilizadas nos ambientes CAD podem ser empregadas para facilitar o processo de representação. Na execução destas ferramentas é necessário informar as dimensões do objeto. Existem duas maneiras de inserir informações sobre o desenho: i) através da barra de tarefas do aplicativo ou ii) carregar os dados através de uma rotina.

No PCEB em estudo as peças e as chapas fazem parte do desenho. Para representar estes componentes uma rotina busca as informações contidas em um relatório (Tabela 3). Este relatório, por exemplo, pode ser um pedido para confeccionar um determinado produto (móvel).

Através das informações do relatório e com o emprego da ferramenta de desenho tipo retângulo, as peças são desenhadas. Uma maneira de organizar o desenho na interface do CAD é desenhando as peças sequencialmente, neste caso é necessário capturar um ponto na tela e desenhar a primeira peça. Neste ponto adiciona-se o comprimento da peça e se encontra a posição de desenho da peça seguinte, e assim sucessivamente enquanto todas as peças sejam desenhadas. A Figura 23 ilustra o desenho ordenado das peças.

Tabela 3 – Modelo de relatório de corte.

Peça (i)	Largura (li)	Comprimento (ci)	Demanda (di)
1	970	570	2
2	700	75	1
3	700	212	3
4	700	163	2

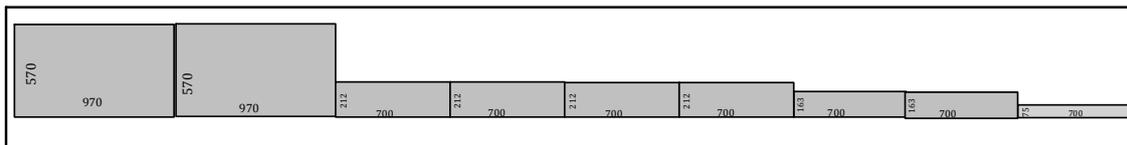


Figura 23 – Representação gráfica das peças.

Uma etapa que antecede a representação geométrica das peças é a criação de camadas. Este recurso possibilita que os componentes do problema sejam representados em camadas distintas, facilitando a seleção e busca de dados específicos. Neste estudo, o desenho gráfico de todas as peças ocupa uma camada, e o desenho da chapa ocupa uma segunda camada. Para utilizar este recurso em ambiente CAD basta executar a respectiva ferramenta e adicionar quantas camadas forem necessárias. Na figura 24 a representação das peças ocupa uma camada denominada demanda e a chapa esta representada na camada arranjo 1.

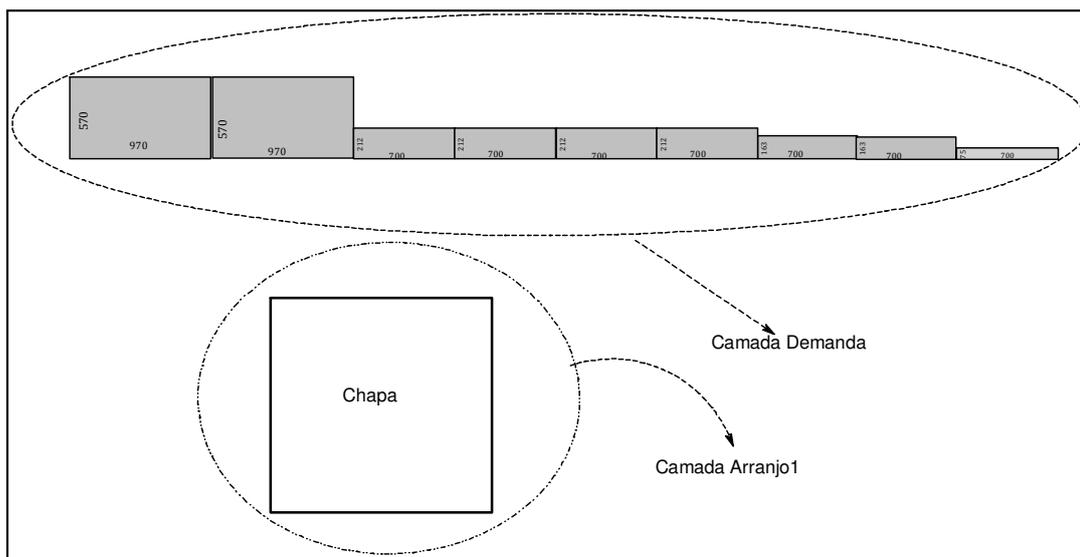


Figura 24 – Camada Demanda e Camada Arranjo 1.

Em ambiente CAD todo desenho é composto por um conjunto de pontos, e cada ponto é formado pelos valores de X e Y que correspondem aos eixos coordenados. Desse modo, é possível calcular largura e o comprimento das entidades gráficas utilizando esta lista de pontos. Entretanto, para o cálculo da área de um desenho os aplicativos CAD disponibilizam um recurso próprio, bastando informar a entidade e o valor da área é calculado. Através da lista de pontos também é possível encontrar o ponto de referência de um determinado desenho, o ponto de referência de cada peça é o ponto formado pelos menores valores de x e y.

3.5.1 Posicionamento das Peças

Para formar o arranjo de corte é necessário posicionar as peças no interior da figura geométrica que representa a chapa. Para este procedimento os aplicativos CAD disponibilizam um recurso de manipulação geométrica para mover os componentes do desenho. Este recurso é processado através dos seguintes parâmetros: entidade que se deseja mover (peça), o ponto de referência desta entidade (definido anteriormente) e o ponto de destino (local onde a peça deverá ser posicionada, neste caso, é um ponto da respectiva chapa).

A lista de pontos para formar o desenho da chapa é inicialmente especificada por uma lista de pontos, como por exemplo: [(10,15); (10,25); (17,15); (17,25)], onde o primeiro valor representa o eixo coordenado X e o segundo valor o eixo coordenado Y. O ponto de destino é encontrado através desta lista de pontos. As seguintes situações podem ocorrer na localização do ponto de destino:

- Quando não existem peças posicionadas na chapa, neste caso, o local de destino terá como critério o ponto que apresentar o menor valor da coordenada x e o menor valor da coordenada y, no exemplo da lista de pontos anterior o ponto de destino será o ponto (10,15).
- Quando uma peça é inserida a lista de pontos da chapa é atualizada. Neste caso o ponto de destino será o ponto que apresentar menor valor de y composto pelo menor valor de x, para isto, deve ser aplicada uma função que ordena os pontos da chapa em ordem crescente levando em consideração somente o valor de y, por exemplo: [(12,15); (17, 15); (10, 25); (17, 25)], se os valores de y forem iguais, os valores de x também são comparados. O ponto que apresentar menor valor de x é considerado o ponto de destino, neste caso (12,15) (Figura 25).

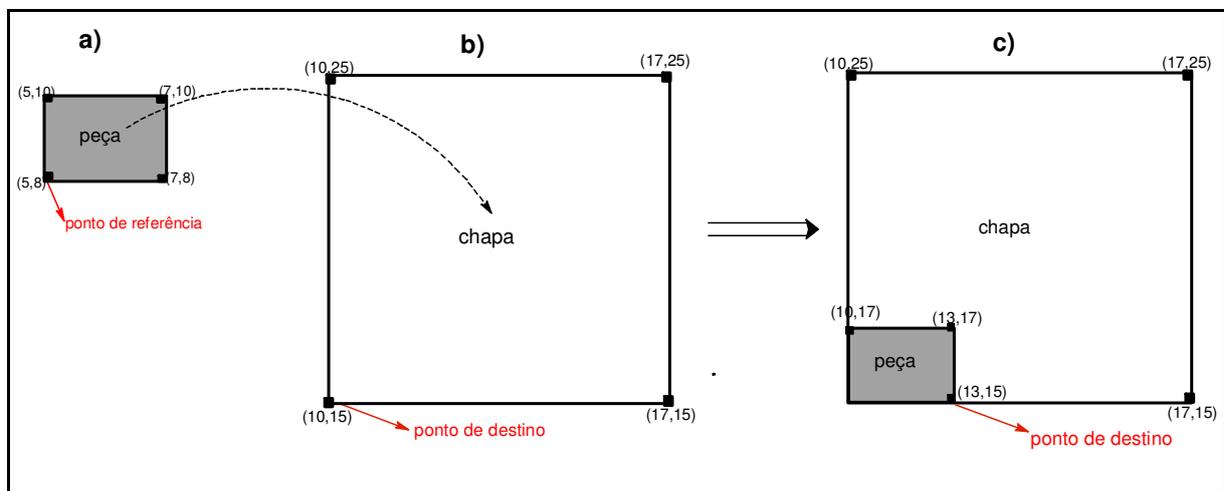


Figura 25 – (a) peça a ser posicionada e o seu ponto de referências; (b) Chapa e o ponto destino; (c) a chapa com a peça e o próximo ponto de destino.

3.5.2 Orientação das peças

Em alguns casos é necessário alterar a orientação das peças para otimizar o espaço da chapa. Os aplicativos CAD disponibilizam de uma ferramenta que executa rotações. No emprego deste recurso é necessário informar os seguintes dados: a entidade que se deseja alterar a orientação (peça), o ponto de referência desta entidade e o grau de rotação, podendo ser um valor fixo, como 90°, ou o valor pode variar conforme a situação. A Figura 26 apresenta a alteração de orientação de uma peça com giro de 90°.

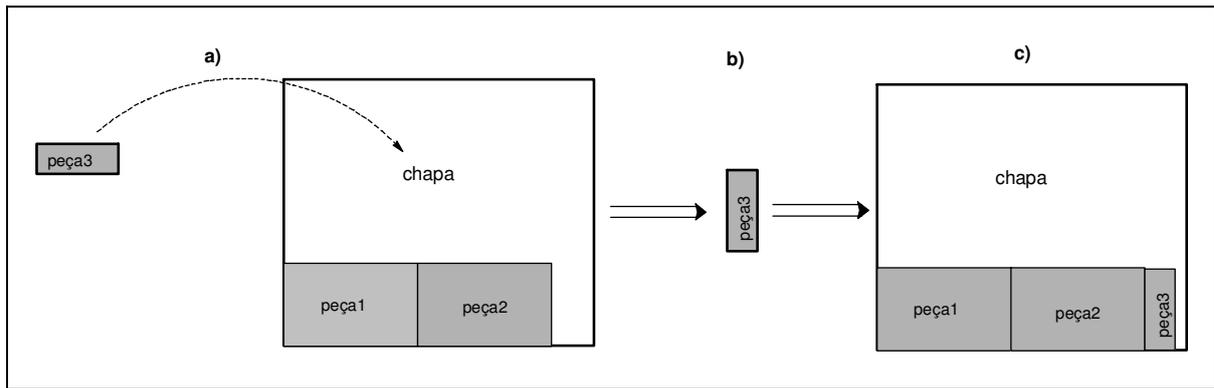


Figura 26 – (a) peça que será posicionada na chapa; (b) peça rotacionada 90° (c) posicionamento da peça na chapa

3.6 Técnica Artesanal

Em grande parte das indústrias de pequeno porte o processo de corte não é automatizado. Neste caso, algum método é adotado para efetuar os cortes com o objetivo de minimizar o desperdício. Por exemplo, a indústria observada o planejamento dos cortes inicia-se a partir do relatório de corte (Tabela 4). Este relatório é repassado para um operário que faz a análise dos cortes, as peças que apresentarem maior área são as primeiras a serem produzidas, conforme ilustra a Figura 27 onde a peça de dimensões 80 x 25 é a primeira a ser cortada. Após a produção das primeiras peças é analisado a dimensão dos retalhos que restaram dos cortes anteriores, então o operário busca no relatório peças que se encaixem nas dimensões dos retalhos. Caso o retalho não possa ser aproveitado na produção de determinado produto é guardado para ser aproveitado em outra situação. Os retalhos são analisados antes de processar os cortes.

Tabela 4 – Relatório de corte.

Peça (i)	Largura (li)	Comprimento (ci)	Demanda (di)
1	80	25	2
2	38	15	1
3	7	18	3

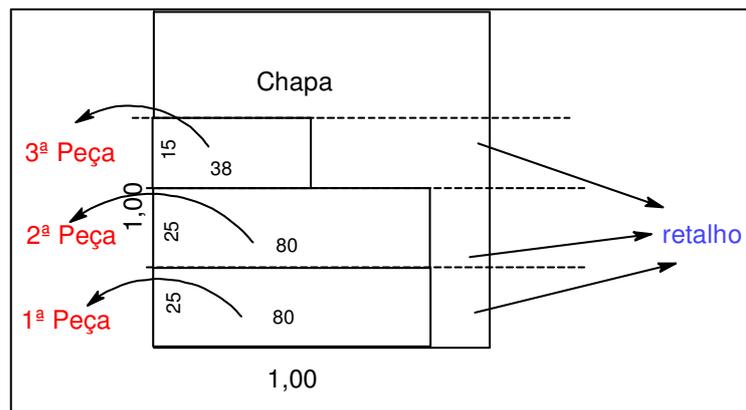


Figura 27 – Demonstração dos cortes para a técnica artesanal.

4. DESENVOLVIMENTO

A metodologia apresentada neste trabalho foi implementada no software AutoCAD (Autodesk, 2010), empregando os recursos de desenho e a linguagem de programação AutoLisp disponíveis neste software. A AutoLISP deriva-se da linguagem computacional LISP (RUSCHEL, 2000), que é a segunda mais antiga linguagem de programação de alto nível ainda usada por computadores modernos. A LISP permite programação concisa para problemas complexos. O AutoLISP é um compilador interno do *software* AutoCAD.

O aplicativo desenvolvido empregou a técnica GRASP para definir os padrões de cortes enquanto que, a representação gráfica e a manipulações dos componentes do problema foram utilizadas as ferramentas do sistema CAD. Além das funções convencionais de qualquer linguagem de programação, o AutoLISP permite também a utilização de comandos de desenho do CAD, bem como a interpretação de dados de entidades geométricas.

O aplicativo desenvolvido inicialmente realiza a leitura do relatório de corte desenhando as peças em forma seqüencial, a seguir o algoritmo começa a processar as entidades do desenho. As entidades desenhadas em *software* CAD como: linhas, arcos retângulos e círculos apresentam um banco de dados contendo as propriedades da respectiva entidade (Figura 28). O algoritmo desenvolvido acessa este banco de dados através de uma rotina denominada *Propri_peça*. Na implementação as seguintes informações foram empregadas: o identificador da entidade no desenho, o tipo da entidade (*line*, *lwpolyline*, *circle*, *3Dface*), o nome da camada (*layer*) e os pontos que formam o desenho (todas as informações que iniciam com o número dez são os pontos que formam o desenho da entidade).

```
((-1 . <Entity name: 7ef03368>) ;nome da entidade
(0 . "LWPOLYLINE") ;tipo da entidade
(330 . <Entity name: 7ef01cf8>) ;marcador de subdivisão de classe
(5 . "1AD") ;
(100 . "AcDbEntity") ;
(67 . 0) ;
(410 . "Model") ;
(8 . "Demanda") ;nome do layer
(100 . "AcDbPolyline") ;classe da entidade
(10 453.472 1485.99) ;ponto inferior esquerdo
(10 807.945 1485.99) ;ponto inferior direito
(10 807.945 1241.75) ;ponto superior direito
(10 453.472 1241.75) ;ponto superior esquerdo
(210 0.0 0.0 1.0)
)
```

Figura 28 – Dados de uma entidade CAD acessados através da rotina *Propri_peça*.

4.1 Implementação do Algoritmo GRASP – PCEB

A partir da representação das peças a algoritmo calcula suas dimensões. O procedimento desenvolvido para calcular as dimensões de todas as peças que compõem o problema é apresentado na Figura 29. Na linha 1, é inicializada a estrutura que armazena os dados (*Lista_peças*). Na linha 2 são inicializada as variáveis *i* – destinada a contar o número de iterações, *ci* – armazena o comprimento, *li* – armazena a largura, *a* – armazena a área e *pto_ref* – guarda o ponto de referência das peças e na linha 3 são inicializadas as variáveis auxiliares. Na linha 5, a estrutura *Lista_ent* receberá todas as peças da camada demanda. O laço principal é compreendido entre as linhas 6 e 17. Neste laço a variável *pi* recebe a peça selecionada na *Lista_ent*, a seguir, a rotina *Propri_peças* acessa o banco de dados da peça corrente e retorna os pontos que formam a mesma, no caso *lista_ptos_peça* (linhas 7 e 8). A partir desta lista de pontos é definido os valores de *ci*, *li* e o *pto_ref*. O valor da variável *a* é encontrado através de uma função área (linhas 9 a 13). A *Lista_peça* armazena todos os valores encontrados (linha 15). O valor de *i* é incrementado enquanto for menor que a quantidade de elementos na *lista_ent*.

	Função Calcula_dimensões_peça
1	<i>Lista_peças</i> ← {};
2	<i>i</i> , <i>ci</i> , <i>li</i> , <i>a</i> , <i>pto_ref</i> : ← 0;
3	<i>x1</i> , <i>x2</i> , <i>y1</i> , <i>y2</i> : ← 0;
5	<i>Lista_ent</i> ← todas as entidades ∈ ao layer “demanda”
6	Enquanto <i>Lista_ent</i> <= <i>i</i> faça
7	<i>pi</i> ∈ <i>Lista_peça</i>
8	Propri_peça (<i>pi</i> :integer, <i>lista_pto_peça</i> : vetor)
9	<i>x1</i> e <i>x2</i> ← pontos com valor de X equivalentes
10	<i>ci</i> ← (- <i>y1</i> <i>y2</i>)
11	<i>y1</i> e <i>y2</i> ← pontos com valor de Y equivalentes
12	<i>li</i> ← (- <i>x1</i> <i>x2</i>)
13	<i>a</i> ← (command “area” <i>px</i>)
14	<i>pto_refe</i> ← pontos com menor valor de (XY)
15	<i>Lista_peça</i> ← (<i>pi</i> ; <i>ci</i> ; <i>li</i> ; <i>a</i> ; <i>pto_refe</i>)
16	<i>i</i> ← <i>i</i> + 1
17	Fim_Enquanto
18	Retorna (<i>Lista_peça</i>)

Figura 29 – Algoritmo para calcular as dimensões das peças.

As dimensões da chapa são informadas pelo usuário. Esta situação possibilita que vários tamanhos de chapas sejam empregadas, inclusive as sobras de matéria-prima. O pseudocódigo da função empregada para desenhar as chapas é apresentado na Figura 30.

Função Desenha_Chapa;	
1	Gerar camada;
2	Solicitar e armazenar o comprimento da chapa
3	Solicitar e armazenar a largura da chapa Executar Função de desenho CAD
5	Fim_Função Desenha_chapa

Figura 30 – Função empregada para gerar as chapas.

O algoritmo proposto permite alterar a orientação das peças no momento do posicionamento para melhor aproveitar um determinado espaço, porém alguns processos produtivos industriais podem impor restrições, como a impossibilidade de sofrer rotação, permitindo somente serem alocadas seguindo uma orientação específica. Por exemplo, nas indústrias de móveis esta alteração está atrelada ao tipo de chapa empregado. As chapas de cor uniforme e lisa não apresentam restrição quanto à orientação, entretanto as chapas que apresentam características intrínsecas da madeira devem ser posicionadas de acordo com o desenho ou posição do veio (Figura 31).



Figura 31 – Exemplo de chapas que restringem a rotação.

Assim, no algoritmo GRASP – PCEB o procedimento restrição de veio permite o usuário indicar quais são as peças que não poderão sofrer alteração de orientação. A Figura 32 apresenta a função verificar a restrição de veio. Na Linha 1, uma variável auxiliar é inicializada. Nas linhas 2 a 5, um laço é executado e as peças com restrição de veio serão indicadas. Esta informação é armazenada na *Lista_peças*. Este laço é repetido enquanto o usuário não informar a condição de saída.

Função verifica_restricção_veio (<i>lista_peças</i>)	
1	<i>var</i> ← 0
2	Enquanto (<i>var</i> = 0) faça
3	Informe as peças ∈ <i>lista_peças</i> com restrição de veio ou digite 1 para sair
4	<i>var</i> ← tecla pressionada
5	Fim_Enquanto
6	Fim_função (<i>Lista_peças</i>)

Figura 32- Função verifica restrição quanto ao veio.

Na fase de construção do algoritmo faixas guilhotinadas viáveis são construídas, iterativamente, analisando uma peça de cada vez. Neste processo dois elementos são fundamentais: uma lista LRC, constituídas pelas próximas peças candidatas a determinar uma faixa; uma função gulosa (*fgul*) que determina as peças *pk* que irão compor a LRC. A peça resultante da escolha aleatória na LRC denomina-se *pk*. A *pk* define uma faixa guilhotinada no sentido horizontal. O algoritmo construtivo é detalhado na Figura 33. Na linha 1, os dados da instância são processados e as peças são representadas graficamente. Na linha 2, a função *Calcula_dimensões_peça* é executada para calcular o comprimento, a largura e área das peças. Nas linhas 3 a 5, são inicializados os seguintes componentes: a estrutura *Solução_inicial*, a variável *iter*, a estrutura auxiliar *S*, a *fav* (função de avaliação) e a LRC. O número máximo de elementos que irão compor LRC é determinado pelo parâmetro *Nº de Elementos* respeitando o critério guloso de maior área. Uma nova camada é gerada e uma chapa é desenhada de acordo com as dimensões informadas pelo usuário (linha 6 e 7). A função restrição de veio é executada para que o usuário possa informar as peças que não podem sofrer rotação (linha 8).

O laço principal é compreendido entre as linhas 9 e 31. Este laço é repetido até que três condições sejam atendidas: o valor de *itermax* (previamente estabelecido) for menor que *iter*, enquanto o comprimento da chapa *C* for maior que zero e enquanto a *Lista_peça* for diferente de zero (linha 9). Uma peça (*pk*) é selecionada aleatoriamente da LRC (linha 10). Na linha 11, é verificado se a peça selecionada não excede a largura da chapa. A seguir a peça é posicionada na chapa de acordo com a técnica de posicionamento *Bottom_left*. As dimensões da faixa são atualizadas (linha 13 á 15). Outro laço é aberto com o objetivo de preencher os espaços na faixa incumbente, sendo executado enquanto as seguintes condições forem verdadeiras: quantidade de elementos na *Lista_peças* menor que a variável contadora (*cont*) e a largura da faixa maior que zero (linha 17). Na linha 18, a *fgul* é adaptada de acordo com o comprimento da faixa, selecionando peças que apresentem o mesmo comprimento ou comprimento inferior que *FC*. Mesmo que a peça não exceda o comprimento da faixa é analisada a possibilidade de rotação com o objetivo de diminuir a de comprimento em relação a última peça inserida (linha 19). Na linha 20 a *pk* sofrerá rotação de 90° caso sua largura seja maior que a largura da faixa. Se a peça apresentar largura inferior ou igual a da faixa a peça corrente é posicionada na chapa, e as variáveis *FL*, *fav* e *cont* são atualizadas (linhas 23 a 27). Na linha 30 a 32 é comparado o comprimento da peça atual com o ultimo comprimento inserido na chapa, se existir diferença entre eles esta diferença é armazenada no vetor *Lista_sobra*. Se o valor da *fav* corrente for maior que o valor *fav* anterior então a *Solução* é atualizada (linha 36 e 37).

```

Função_Construtiva
1  Leia e processe os dados da instância
2  Processe a função Calcula_dimensões_peça (Lista_peças: vetor);
3  Solução_inicial ← {}; iter ← 0; FH ← {}
4  Fgul ← {}; S ← {}; fav ← {};
5  LRC ← Lista_peças
6  Gerar nova camada (layer);
7  Executa Função Desenha_Chapa (C, L);
8  Processa Função Verifica_restrição_veio;
9      Enquanto (itermax < iter) E (C > 0) E (Lista_peça # 0) faça
10     Escolher aleatoriamente, (fgul) uma peça (pk) na LRC
11     Se (li <= L) então
12         Posicionar pk na chapa
13         FL = L
14         FC = ci
15     Fim_se
16     Enquanto (cont <= Lista_peça) E (FL > 0) faça
17         Adaptar Fgul de acordo com FC
18         Selecionar pk
19         Se ci # FC e li diminuir a diferença entre o FC então Rotacionar 90° pk
Fim_se
20     Se li >= FL e ci <- FC
21         Rotacionar pk;
22     Fim_Se
23     Se li <= FL
24         Posicionar pk na chapa
25         Atualizar pontos da chapa
26         Remove pk da Lista_peças
27         FL = FL - li
28         fav = fav + a
29         cont = cont + 1
30         Se ci anterior # ci
31             Dif ← ci - ci anterior
32             Lista_Sobra ← dif e pto_ref
33         Fim_Se
34     Fim_Se
35     Fim_Enquanto
36     Se fav corrente > fav então
37         Solução_inicial ← S corrente
38     Fim_se
39     iter = iter + 1
40 Fim_Enquanto
41 Fim_função_construtiva (Solução_inicial)
42

```

Figura 33 – Procedimento Construtivo do algoritmo GRASP-PCEB.

A melhor faixa encontrada na função construtiva é incluída no arranjo de corte. A partir desta faixa, se existir áreas não ocupadas a função melhoramento é executada. Esta função é detalhada na Figura 34. A *Solução_inicial* no caso a faixa é um dado de entrada resultante da função construtiva (linha 1). A largura da sobra, o comprimento da sobra e a menor largura das peças representadas respectivamente por: *lar_sobra*, *com_sobra*, *menor*, são inicializadas (linha

2). Enquanto houver espaços para ser ocupado um *loop* é executado (linhas 3 e 4). As variáveis *larg_sobra* e *com_sobra* são atualizadas conforme valores da *lista_sobra*. A variável contadora é incrementada (linhas 5 a 7). Um laço é repetido enquanto a *larg_sobra* for maior que a *menor_largura* (linha 8). Uma peça é selecionada na *Lista_peças* executando as seguintes comparações: largura da peça é menor que largura da sobra, comprimento da peça é menor que comprimento da sobra se ambas as comparações forem verdadeiras a peça é posicionada no espaço não preenchido corrente. Os pontos da chapa e a largura da sobra são atualizados, ver linhas 9 a 13.

```

Função melhoramento (Solução_inicial)
1   $S \leftarrow Solução\_inicial$ 
2   $lar\_sobra \leftarrow 0; comp\_sobra \leftarrow 0; menor\_l \leftarrow 0;$ 
3  Identifica espaços vazios e armazena na lista_sobra
4  Enquanto (lista_sobra  $\neq 0$ ) E (cont  $\leq$  ) faça
5      Atualizar larg_sobra e comp_sobra
6      Identificar menor_l  $\in$  a Lista_peças
7       $cont = cont + 1$ 
8  Enquanto (larg_sobra  $\geq$  menor_l) faça
9      Selecionar pi (ci,li) da Lista_peças
10     Se (li  $\leq$  larg_sobra) E (ci  $\leq$  comp_sobra) então
11         Posicionar pi no espaço vazio da faixa
12         Atualizar pontos da chapa
13          $larg\_sobra \leftarrow larg\_sobra - li$ 
14     Fim_Se
15 Fim_Enquanto
Fim_Enquanto

```

Figura 34 – Função de melhoramento.

O algoritmo geral produz os arranjos de corte (Figura 35). O laço é repetido até que todas as peças sejam posicionadas (linha 2). Para a representação das chapas em camadas distintas é empregando o recurso *layer* (linha 3). A seguir as dimensões (comprimento e largura) da chapa são calculadas (linha 3 e 4). A Função construtiva é executada resultando uma solução parcial. A função melhoramento é executada a partir dos dados oriundos da função construtiva. Ao final da execução deste algoritmo são produzidos os arranjos de corte.

```

Algoritmo Geral
2  Enquanto (Lista_peças  $\neq 0$ ) faça
3      Gera camada (command “layer” “N” chapax )
4      Calcular dimensões chapa (C, L)
5      Função construtiva (Solução_inicial)
6      Função de melhoramento (Solução_finall)
7  Fim_Enquanto
8  Fim Algoritmo geral (Solução_final)

```

Figura 35 – Algoritmo GRASP –PCEB.

5. RESULTADOS

As ferramentas CAD e GRASP descritas na metodologia podem ser aplicadas a vários segmentos industriais que envolvem o processo de corte. Neste trabalho, testes computacionais foram direcionados para uma indústria moveleira. Essa indústria produz móveis sob medida, no qual, os desenhos das peças do móvel são realizados com auxílio de aplicativos CAD onde também é gerado um relatório de corte contendo as dimensões das peças (Tabela 5). A partir deste relatório o programa implementado GRASP-PCEB produz os arranjos de corte. Uma máquina seccionadora é utilizada para efetivar os cortes do tipo guilhotinado.

Tabela 5 – Dados das instâncias B2p, A3p e R5p.

Instância	Peça (<i>i</i>)	Largura (<i>li</i>)	Compr. (<i>ci</i>)	Demanda (<i>di</i>)
B2p	1	56,4	47,5	1
	2	8	193,6	2
	3	50	58,2	2
	4	48,8	57,9	2
A3p	1	30	116,8	1
	2	30	60	2
	3	28,5	116,4	1
	4	39,1	57,9	3
	5	27,5	77	4
	6	28,5	56,4	1
R4P	1	55,8	249,8	4
	2	55,8	97,5	4
	3	55,8	47,8	2
	4	55,8	49,6	1
	5	54,9	97,5	2
	6	54,9	80	1
	7	54,8	47,8	1
	8	53,9	47,8	5
	9	38,6	15,8	8
	10	15,8	49,9	16
	11	7,8	249,6	2
	12	7,8	48,4	2
	13	48,4	7	5

Os resultados obtidos com o algoritmo foram executados em um microprocessador Celeron-Intel, com 2Gb de RAM e sistema operacional Windows Vista. Os testes com o algoritmo foram realizados através de um grupo de instâncias referentes a diferentes tipos de móveis. A fim de validar o algoritmo proposto, são comparados os valores da função de avaliação gerados pelo programa GRASP – PCEB com os valores apresentados por um *software* de referência, na versão *Trial* (Corte Certo, 2010), aplicando o mesmo conjunto de instâncias. Os valores da função de avaliação também são apresentados para a técnica artesanal.

Três instâncias foram empregadas para realização dos testes, sendo elas: um roupeiro 5 portas (R5p), um armário 3 portas (A3p) e um balcão 2 portas (B2p). Cada bloco, na Tabela 5, representa uma instância, onde serão exibidas as seguintes informações: índices (i), comprimento (li), largura (wi) e demanda (bi) das peças, respectivamente. As peças destas instâncias empregam matéria-prima com espessura de 8mm. Assim nos arranjos de corte as peças que formam os fundos dos móveis não foram inclusas pois empregam matéria-prima com espessura igual a 6mm. Para efeitos de comparação, toda a sobra foi designada como desperdício.

Para a execução de um teste realizado a partir dos dados da instância A3p, empregou-se a restrição quanto a rotação das peças. Para os demais testes é considerada a rotação de 90° sobre as peças.

5.1 Instância 1

A primeira instância está composta por peças do móvel roupeiro de 5 portas. A Figura 36 apresenta este móvel. Os resultados obtidos para a instância R5p após a execução dos programas GRASP –PCEB e de Referência são apresentados na Tabela 6. Para este exemplar é apresentado as chapas empregadas identificadas pelo campo Chapas. Para todas as chapas são informados o valores individuais obtido pela função de avaliação (Fav) e pelo do cálculo de desperdício ($Desp.$)

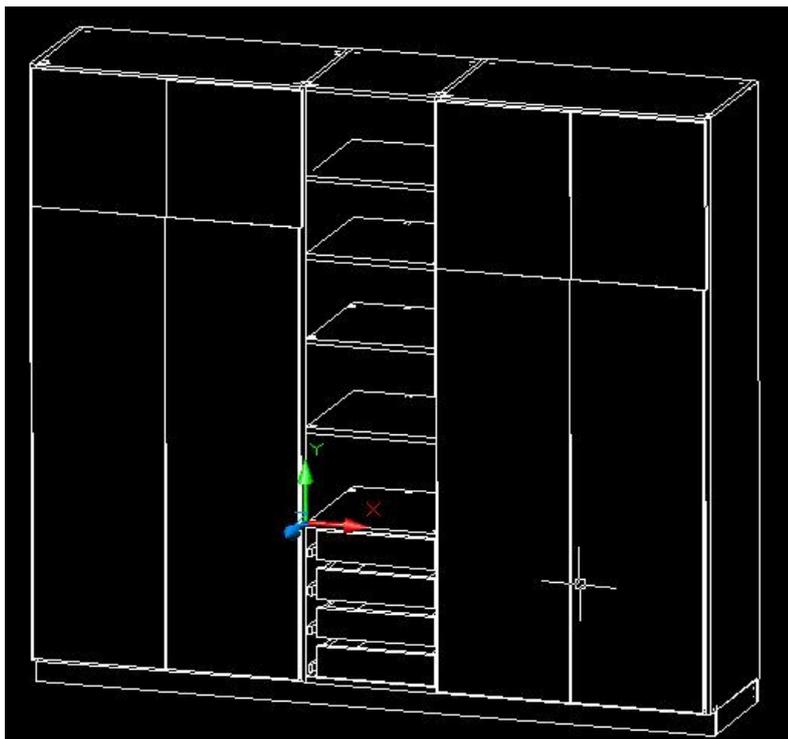


Figura 36 – Móvel armário

Tabela 6 – Resultado para instância R5p.

GRASP-PCEB			Referência			Artesanal		
<i>Chapas</i>	<i>Fav (%)</i>	<i>Desp. (%)</i>	<i>Chapas</i>	<i>Fav (%)</i>	<i>Desp. (%)</i>	<i>Chapas</i>	<i>fav</i>	<i>Desp.</i>
1	96,43	3,5	1	93,79	6,21	1	83,1	16,9
2	91,85	8,15	2	85,23	14,77	2	81,4	18,6
3	89,95	10,05	3	82,03	17,97	3	90,1	9,9
						4	-	-

A instância R5p empregou chapas com dimensão fixa de $C = 1,83$ cm e $L = 2,75$ cm. Para atender toda a demanda de peças os programas GRASP-PCEB e de Referência empregaram três chapas e a técnica artesanal ocupou quatro chapas. O valor médio da *fav* das soluções finais foi de 93%. O valor encontrado pelo cálculo de desperdício foi inferior a 11%. Analisando os resultados obtidos, o programa GRASP-PCEB apresentou resultado superior quando comparados ao Programa de Referência que obteve média da *fav* de 87% e valor de desperdício médio de 12,98%. O método artesanal também foi aplicado nesta instância, resultando um valor médio na *fav* de 85%. As Figuras 37, 38 e 39 apresentam os arranjos de corte gerados pelos Programas GRASP-PCEB, de Referência e pela técnica artesanal, respectivamente. Considerando os resultados apresentados pelo programa GRASP-PCEB desta instância, observa-se que quanto maior a diversidade de peças, melhor é o aproveitamento dos espaços na chapa.

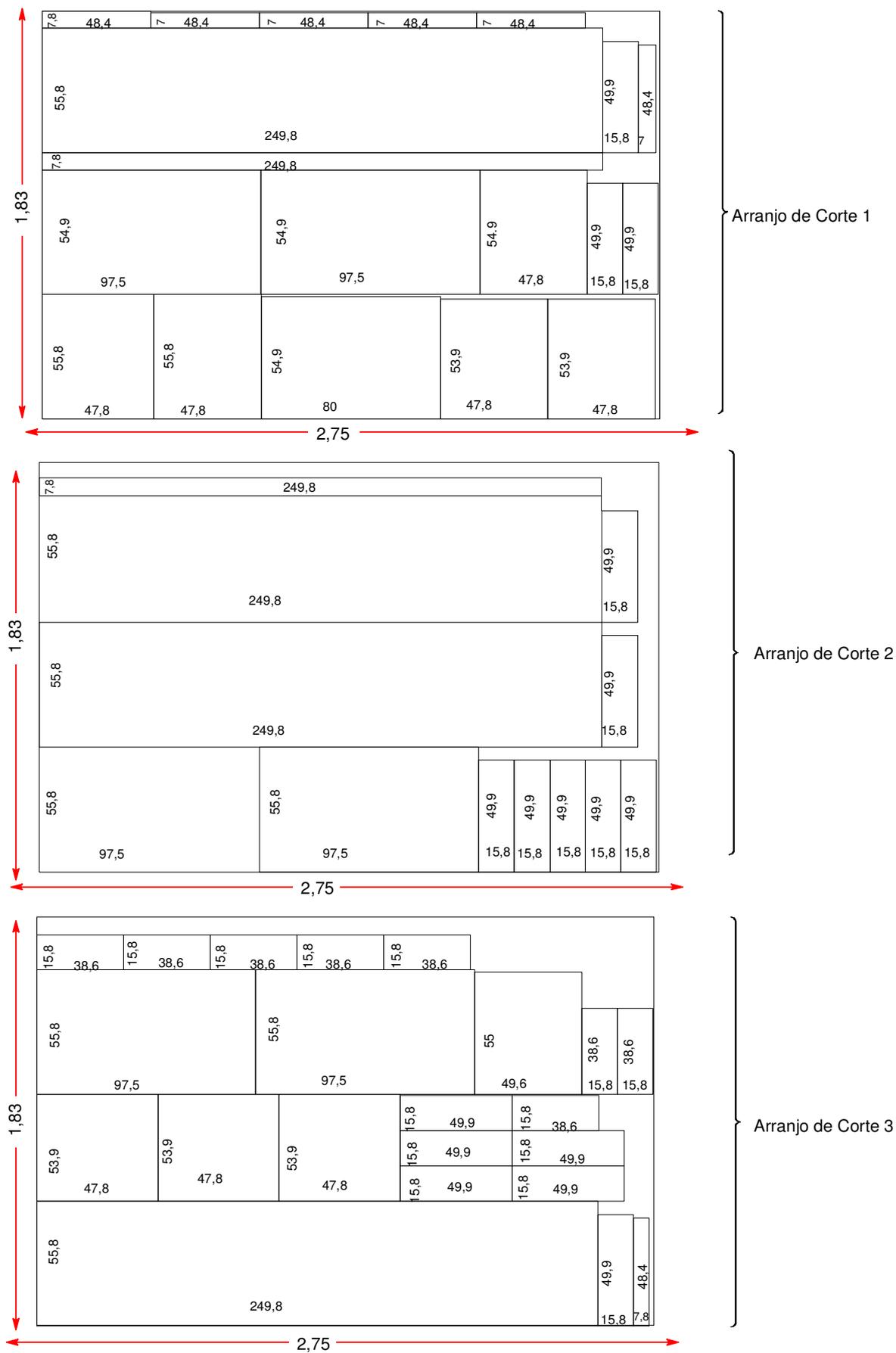


Figura 37 – Arranjos de corte gerados pelo programa GRASP-PCEB

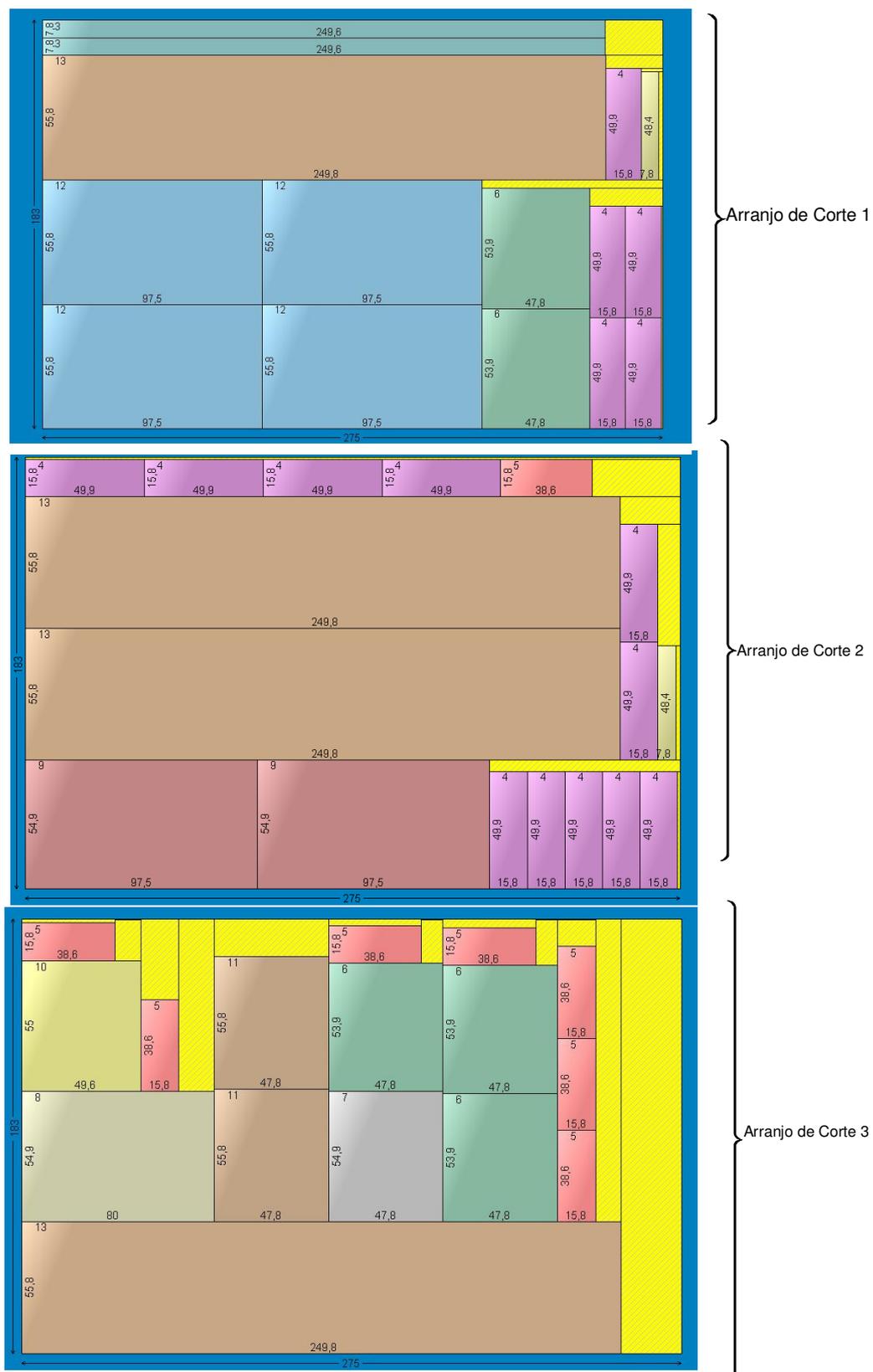


Figura 38 – Arranjos de corte gerados pelo Programa de Referência

No primeiro teste, para a instância A3p, consideraram-se chapas com dimensão de $C = 2,75$ cm e $L = 1,20$ cm. Na execução do programa GRASP-PCEB assim como na execução do Programa de Referência foi necessário apenas uma chapa para arranjar as peças. Comparando os resultados de *fav* e de desperdício obtidos pelos programas constatou-se que: o valor da *fav* quanto de desperdício se equivaleram, obtendo 82,5% e 18%, respectivamente. Em relação ao número de estágio, os arranjos de corte gerados necessitam três estágios para produzir as peças. Analisando o arranjo de corte gerado por GRASP-PCEB (Figura 41) observa-se que a área não ocupada apresenta-se agrupada em uma única região, conseqüentemente, aumentando a possibilidade de aproveitamento. A Figura 42 apresenta o arranjo de corte gerado pelo Programa de Referência.

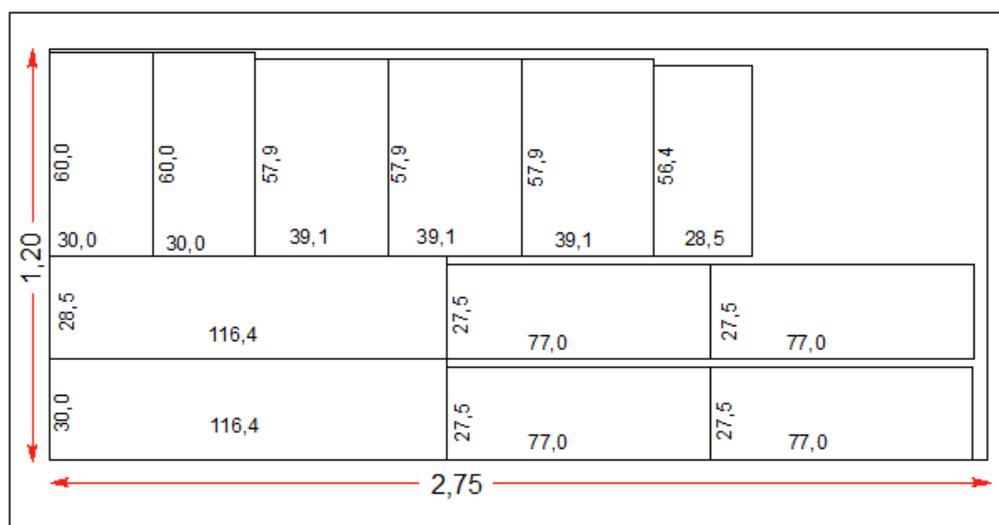


Figura 41 – Arranjo de corte da instância A3p gerado pelo GRASP

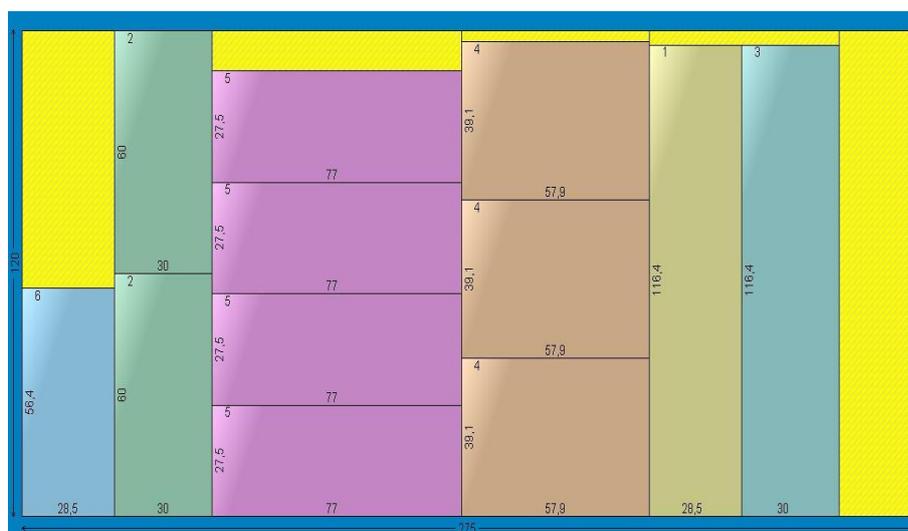


Figura 42 - Arranjo de corte da instância A3p gerado Programa de Referência

No segundo teste para instância A3p empregou-se chapas com dimensão de $C = 1,50$ cm e $L = 1,50$ cm. A Tabela 7 apresenta os resultados observados para a *fav* e o desperdício.

Tabela 7 – Resultados da instância A3p.

GRASP-PCEB			Referência			Artesanal		
<i>Chapas</i>	<i>Fav (%)</i>	<i>Desp. (%)</i>	<i>Chapas</i>	<i>Fav (%)</i>	<i>Desp. (%)</i>	<i>Chapas</i>	<i>Fav (%)</i>	<i>Desp. (%)</i>
1	87,1	23,9	1	93,03	17	1	75,5	24,5
2	-	-	2	-	-	2	-	-

Na execução deste teste os programas utilizaram duas chapas para arranjar as peças. Os resultados da *fav* e de desperdício para segunda chapa não foram calculados, pois as peças não preencheram a chapa. Os arranjos de corte do programa GRASP-PCEB e de Referência são apresentados nas Figuras 43 e 44, respectivamente. No primeiro arranjos de corte gerado pelo Programa de Referência foi possível observar um melhor aproveitamento no espaço da chapa quando comparado ao arranjo produzido por GRASP-PCEB. Neste caso, o programa de referência otimizou o espaço da chapa em 6,9% em relação ao GRASP-PCEB. Neste teste os resultados para técnica artesanal foram inferiores aos resultados obtidos pelos outros programas, a Figura 45 apresenta o arranjo de corte quando empregado esta técnica.

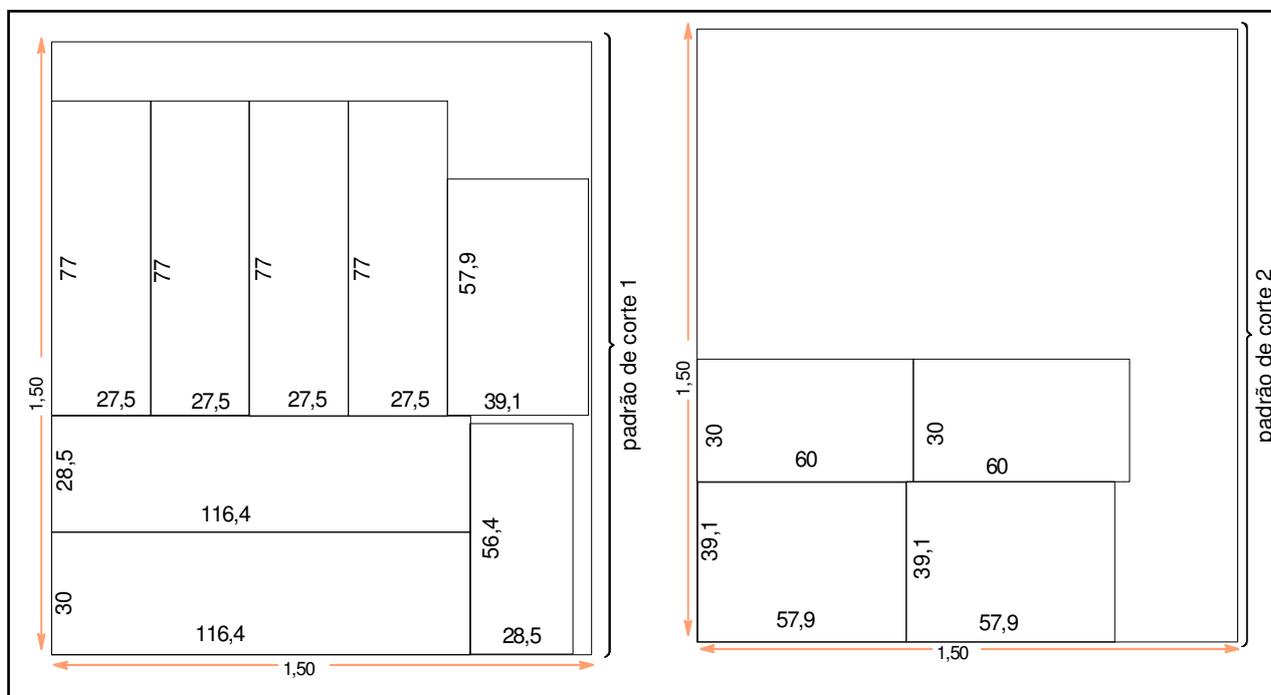


Figura 43 - Arranjo gerado pelo programa GRASP- PCEB

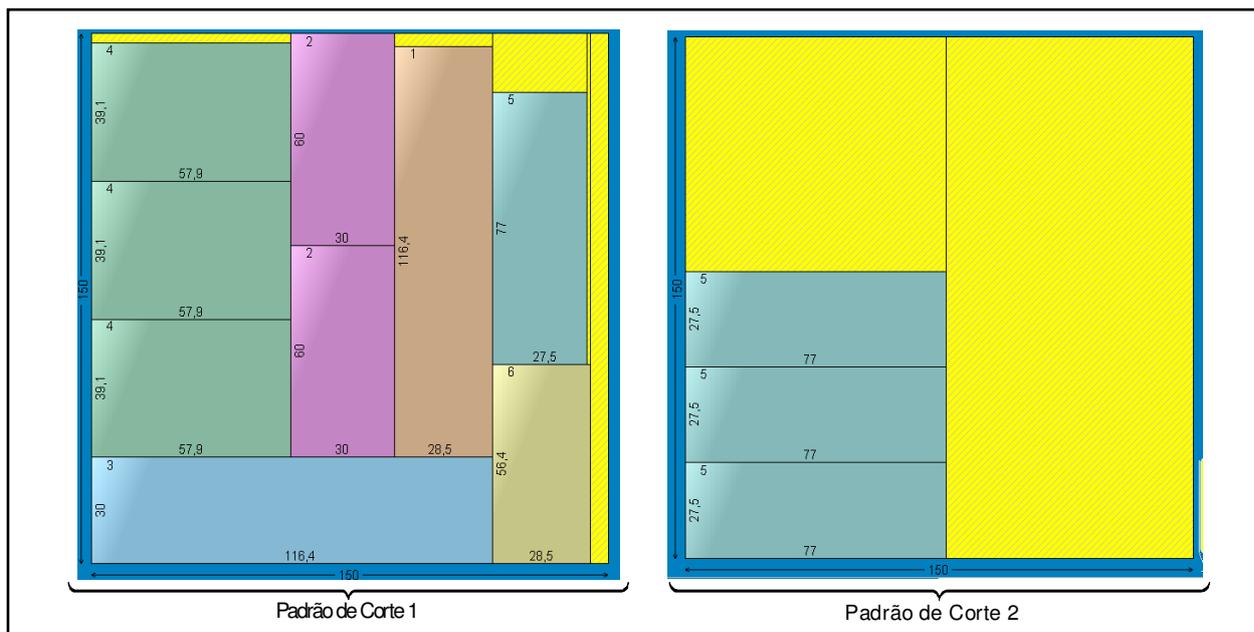


Figura 44 – Arranjo gerado pelo Programa de Referência

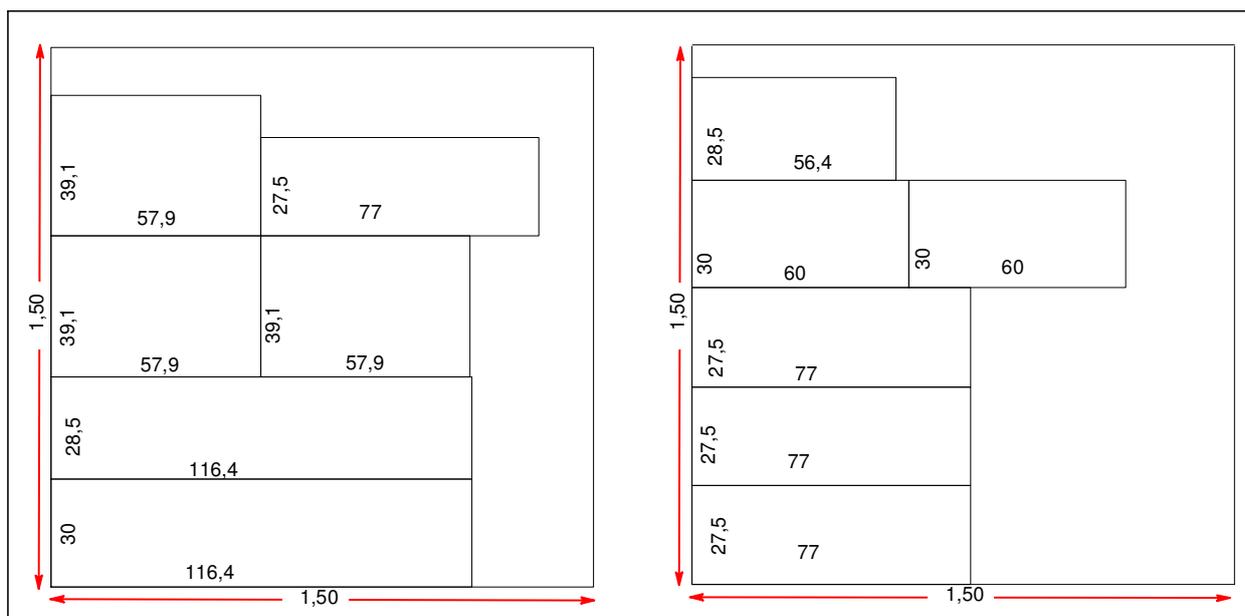


Figura 45 – Arranjo gerado pela Técnica Artesanal

O último teste realizado com a instância A3p restringiu-se a rotação das peças, ou seja, as peças devem ser posicionadas na chapa de acordo com sua orientação. As chapas empregadas apresentam dimensão de $C = 1,50$ cm e $L = 1,50$ cm. Observa-se que o uso desta restrição implica em arranjos de corte inferiores e um maior valor de desperdício, tanto para o arranjo de corte gerado pelo programa GRASP-PCEB (Figura 46) quanto nos gerados pelo Programa de Referência (Figura 47).

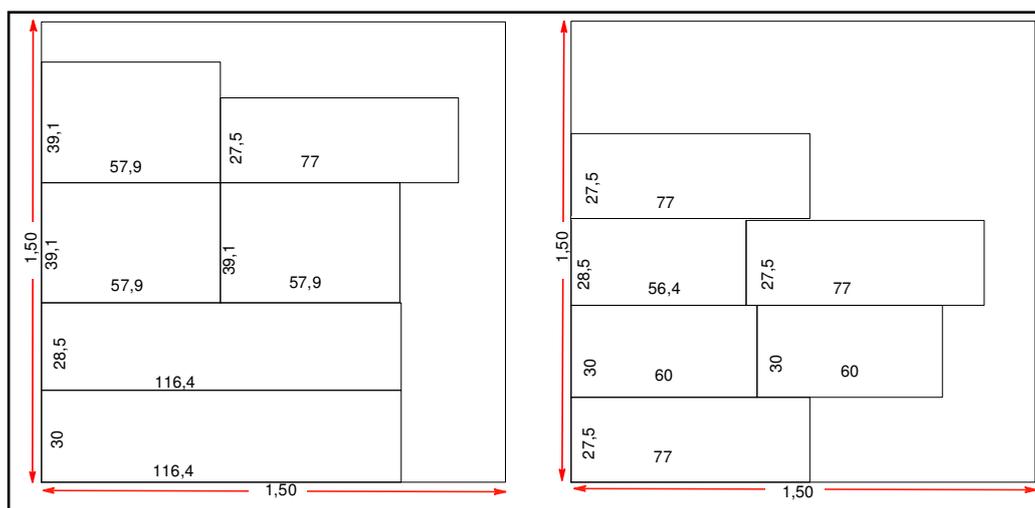


Figura 46 – Arranjos com restrição, gerado pelo Programa GRASP – PCEB.

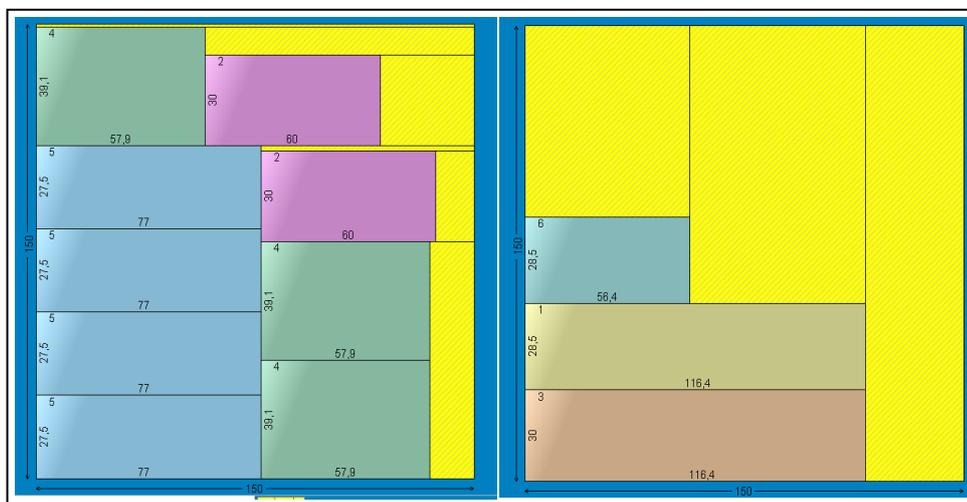


Figura 47 – Arranjos com restrição, gerado pelo Programa de Referência.

5.3 Instância 3

A terceira instância B2p, considerada neste estudo, é composta pelas peças que formam o móvel balcão apresentado na Figura 48. A Tabela 8 apresenta as informações da *fav* e do desperdício obtidos pelos programas GRASP-PCEB e de Referência.

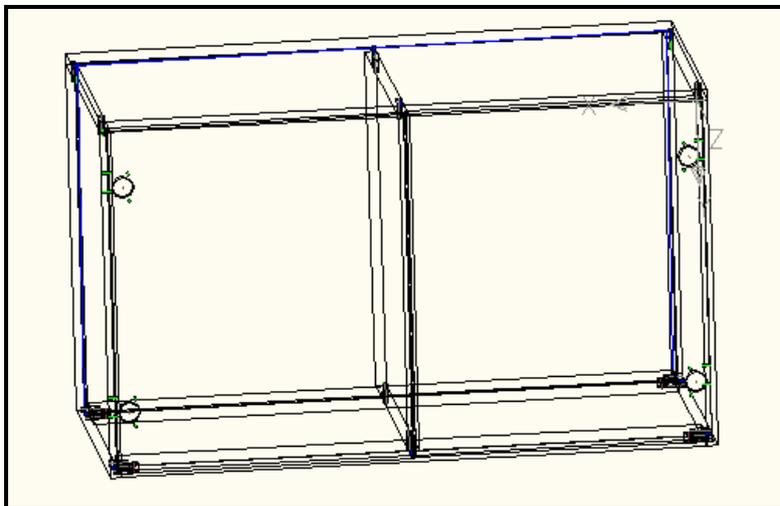


Figura 48 – Representação do móvel Balcão.

Tabela 8 – Resultados da instância B2p.

GRASP-PCEB			Programa de Referência		
<i>Faixa</i>	<i>Fav (%)</i>	<i>Desp. (%)</i>	<i>Faixas</i>	<i>Fav (%)</i>	<i>Desp. (%)</i>
1	92,32	8,68	1	97,72	2,28
2	49,7	50,3	2	77,27	22,2

Nesta situação foi necessária apenas uma chapa para que os programas formassem os arranjos de corte (Figura 49). Porém, o valor de *fav* e de desperdício foi calculada para cada faixa que compõem o arranjo de corte. Nas faixas do arranjo de corte do Programa de Referência os valores de *fav* foram superiores quando comparados aos valores da *fav* de GRASP-PCEB. Além disso, observa-se que o Programa de Referência otimizou o espaço da chapa, sendo que a sobra existente neste arranjo de corte será facilmente aproveitada.

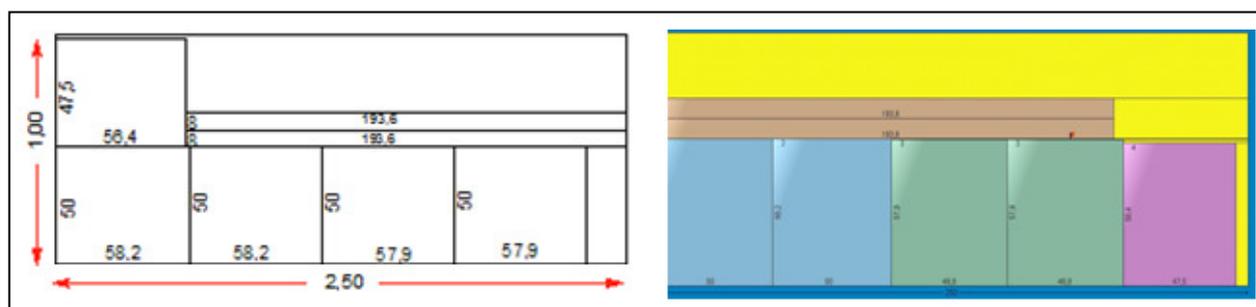


Figura 49 – Arranjo de Corte (a)GRASP-PCEB; (b) de Referência.

5.4 Programa CNC

Um dos objetivos deste trabalho foi propor um modelo de integração CAD/CAM. A partir dos arranjos formados pelo algoritmo GRASP-PCEB, é demonstrada a geração de dados para o ambiente CAM. Estes dados apresentam as seqüências de operações codificadas em linguagem CNC para serem interpretadas pela máquina de corte através de Comandos Numéricos.

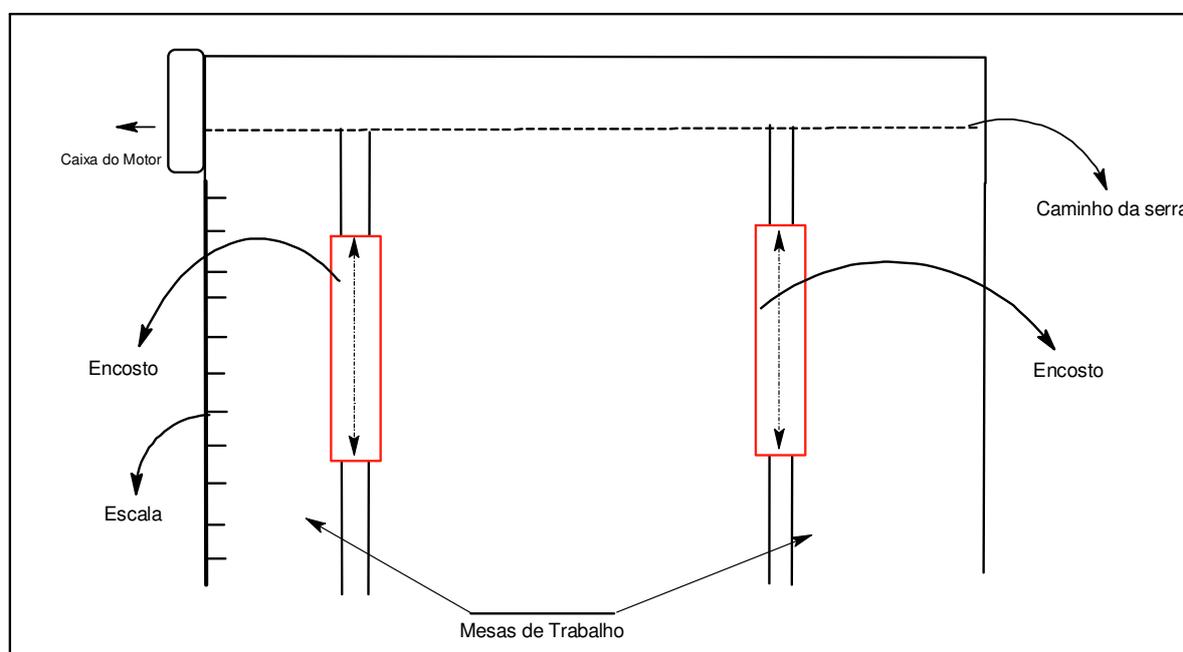
A máquina de corte, utilizada na indústria moveleira, é uma seccionadora composta por uma mesa que contém uma serra circular atrelada a um motor e os encostos que são programados de acordo com as dimensões da chapa (Figura 50).

Os encostos podem sofrer ajustes sempre que existir uma nova chapa a ser inserida. Nesta seccionadora quando o motor é acionado, a serra percorre toda a extensão da mesa, efetivando o corte sobre a chapa (Figura 51). No caso desta máquina os encostos são controlados por códigos CN, ou seja, um programa CNC controla os eixos dos encostos permitindo que os cortes sejam efetuados automaticamente. Atualmente, neste setor da indústria, os operários realizam o processo de carga e remoção das chapas. Quando a alimentação da matéria prima na máquina é o lado oposto do encosto, o programa CNC deve conter diretamente as coordenadas de corte o que ocorre no primeiro estágio. Porém, após o segundo ou terceiro estágio, as dimensões das faixas permitem que a alimentação seja do mesmo lado do encosto. Nessa situação, as coordenadas do programa devem ter como referência o comprimento da faixa.

O programa CN, desenvolvido para a máquina seccionadora em estudo empregou os seguintes comandos: G0 – este comando é necessário para o encosto se movimentar em velocidade rápida, M0 – comando utilizado para pausar o programa, M3 – quando acionado este comando liga a serra de corte, M2 – finaliza o programa, X indica as posições absolutas relacionadas a coordenada X, ou seja, o ponto onde a chapa deverá ser posicionada para iniciar os cortes. No programa CN o ponto de posicionamento é calculado através da seguinte fórmula: $X=(lim - comp)$, onde a variável *lim* indica o ponto na chapa onde a serra iniciará o corte e *comp* representa o comprimento da respectiva chapa.



Figura 50 – Seccionadora da indústria de móveis



Figuras 51 – Componentes da máquina seccionadora

A proposta de programa CN produzirá as peças do arranjo de corte da instância B2p que representa o móvel balcão (Figura 52).

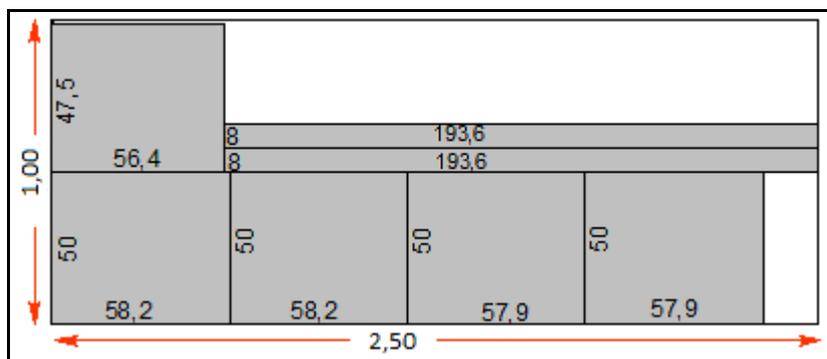


Figura 52 – Arranjo de corte Modelo.

O pseudo-código implementado em linguagem CN para executar os cortes é apresentado na Figura 53. A representação dos primeiros cortes realizados no sentido horizontal é ilustrada na Figura 54a.

Inicialmente, os encostos da máquina seccionadora são ajustados de acordo com o comprimento da chapa (linha 1), o comando M0 coloca o programa em modo de espera e envia uma mensagem ao operador solicitando que a chapa com as dimensões informadas seja posicionada. Neste exemplo, a chapa apresenta 2,50 m de largura e 1,00 m de comprimento (linhas 2 e 3). A seguir os encostos são ajustados com a posição de corte e neste caso a chapa avança 50 cm em relação a serra. Este procedimento posiciona a chapa no ponto ideal para ser efetuado o corte (linha 4) e o comando M3 aciona a serra que percorre toda a extensão da chapa efetivando o primeiro corte (linha 5). Os encostos são ajustados novamente e a chapa avança 47,5 cm em relação a serra. O comando M3 acionará a serra e a chapa sofrerá o segundo corte e entrará em modo de espera (linha 6 a 8). Neste ponto do programa os cortes produziram as faixas restando ainda os cortes internos para produzir as peças.

Após produzir as faixas, os encostos são ajustados de acordo com a próxima chapa (2,50 x 50). O programa mostra as dimensões e a chapa é conduzida na posição de corte (17,8), o comando M3 aciona a serra para efetivar o corte. Neste exemplo, o corte produz uma sobra (linha 9 a 12). Para efetuar os próximos cortes desta chapa os comandos que controlam a máquina seccionadora se repetem pois o objetivo é o mesmo programar os encostos avançando a chapa ate a serra e executar o corte (linha 13 a 20). A Figura 54b ilustra esta etapa.

Na linha 21, o programa pausa e envia uma mensagem para o operador retirar as peças. Os encostos novamente são programados de acordo com as dimensões da próxima chapa (2,50 x 47,5). No próximo ajuste dos encostos a chapa avançará 193,6 cm em relação a serra (linha 22 e

23) a peça resultante deste corte retornará para máquina seccionadora e os encostos ajustados para largura 1,93 e comprimento 47,5 cm (linha 31). Após posicionada a respectiva chapa, os encostos serão programados para avançar 31,5 cm em relação a serra. Finalmente os últimos cortes serão executados produzindo assim todas as peças (linha 32 a 37). O comando M2 encerra o programa (linha 40).

Programa CN			
1	G0 (X + 1,00)	21	M0; remover peças
2	M0	22	X(lim – 2,50)
3	;posicionar a chapa (1,00 x 2,20)	23	;posicionar a chapa (2,50 x 47,5)
4	(X + 50)	24	X(lim – 193,6)
5	M3	25	M3
6	(X + 97,5)	26	X(lim – 2,50)
7	M3	27	M3
8	M0	28	;remover peças
9	(X + 2,50)	29	M0
10	;posicionar a chapa (2,50 x 50)	30	X(lim – 2,50)
11	X(lim – 17,8)	31	;posicionar a chapa (47,5 x 1,93)
12	M3	32	X(lim – 31,5)
13	X(lim – 75,7)	33	M3
14	M3	34	X(lim – 39,5)
15	X(lim – 133,6)	35	M3
16	M3	36	X(lim – 47,5)
17	X(lim – 191,8)	37	M3
18	M3	38	M0
19	X(lim – 2,50)	39	;remover peças
20	M3	40	M2

Figura 53 – Programa CN para o arranjo Modelo.

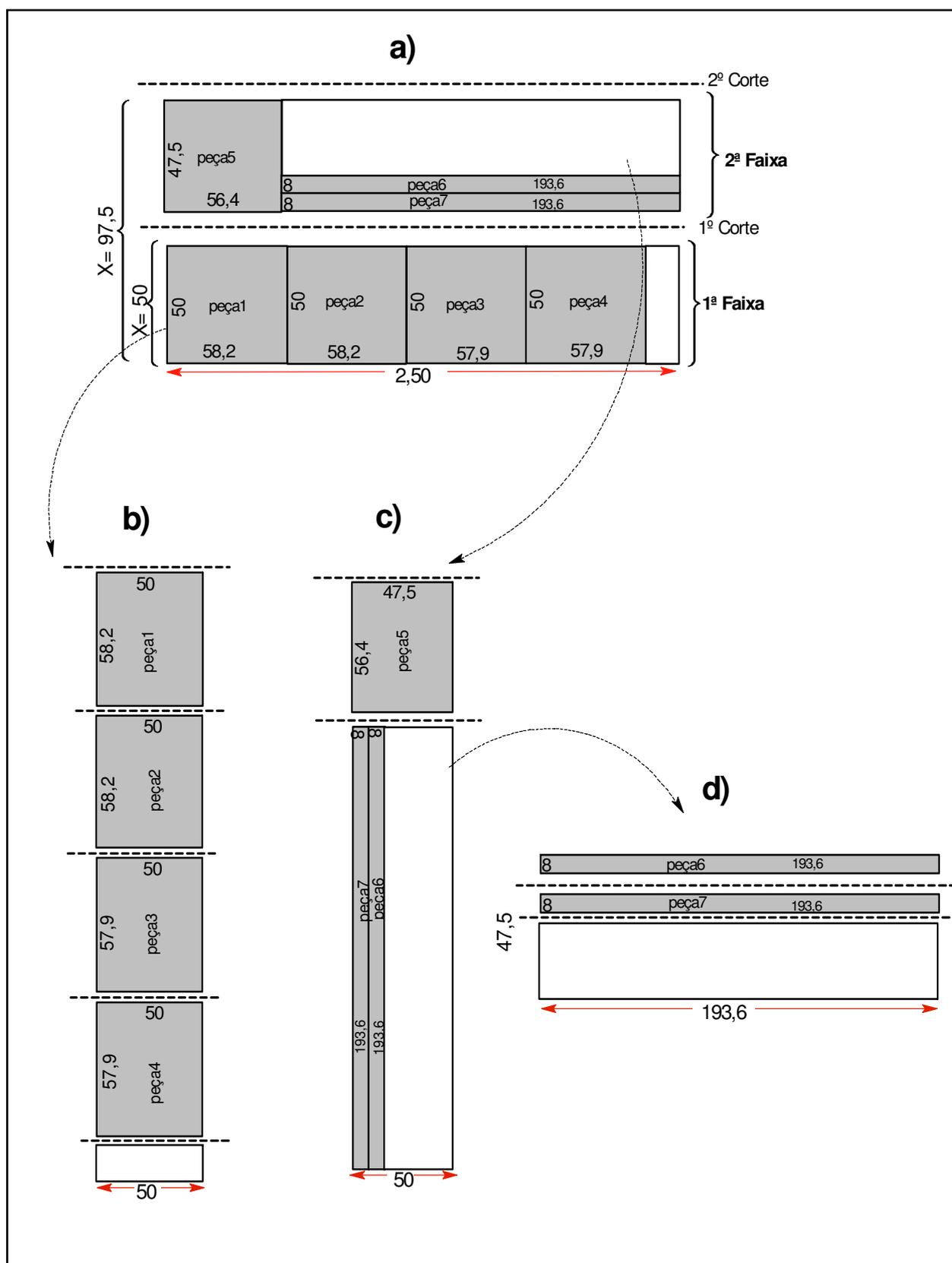


Figura 54 – Processo de corte empregado pelo programa CNC.

6. CONCLUSÃO

A implementação da metodologia proposta visa a resolução do PCEB através da técnica GRASP implementada em ambiente CAD. A partir dos testes envolvendo dados provenientes do setor industrial, é possível concluir que:

- A meta-heurística empregada consegue encontrar soluções viáveis para o problema de otimização combinatória. Porém, observa-se que a formação de arranjos de corte com faixas somente no sentido horizontal privou a geração de arranjos com maior índice de aproveitamento e, conseqüentemente, menor desperdício.
- O desempenho do algoritmo desenvolvido empregando métodos heurísticos e ferramentas CAD foi analisado pela comparação com os resultados obtidos com as soluções geradas por um software comercial especializado. Constatou-se que o programa desenvolvido gera soluções factíveis, entretanto, a qualidade das soluções esta diretamente relacionada com a diversidade de peças demandadas.
- Quando imposta a restrição de rotação de posicionamento nas peças, ocorrem perdas consideráveis tanto no programa GRASP-PCEB quanto no Programa de referência. Porém, a importância desse recurso o torna indispensável para o caso da indústria moveleira.
- Em relação ao emprego de ferramentas CAD nos problemas de otimização combinatória, observa-se uma alternativa viável, quando aplicada às indústrias que trabalham com objetos personalizados, no caso um ramo específico da indústria moveleira. Entretanto, o uso destes recursos, quando aplicados a instâncias consideradas de grande porte, não é indicado, pois tornam o processamento lento, inviabilizando o tempo de resposta.
- A proposta de adequação do modelo a máquinas CNC demonstra que a metodologia é adequada para a implementação de um módulo de geração automática de programas CNC para o caso de desenvolvimento de sistemas CAD/CAM dedicados.

- Propõe-se para trabalhos futuros i) a aplicação de novas meta-heurísticas na resolução do problema estudado; ii) implementação da metodologia proposta em outros setores industriais também pode ser um caminho de continuidade dos presentes esforços de pesquisa nesse tema; iii) implementação do módulo de geração automática de programas CNC; iv) utilização da modelagem computacional exposta neste trabalho para o tratamento de outras variações de Problemas de Corte, tanto unidimensional como bidimensional com formas regulares e irregulares.

7 REFERÊNCIAS BIBLIOGRÁFICAS

- ABIMOVEL – Associação Brasileira de Indústria de Mobiliário. Disponível em <www.abimovel.com.br>. 2010.
- ALVAREZ-VALDES, R.; PARRENO, F.; TAMARIT, J.M. **A GRASP algorithm for constrained two-dimensional non-guillotine cutting problems**, Journal of the Operational Research Society 56, 414-425. 2005.
- ARAUJO, B.C.O. **Problema de corte e empacotamento tridimensional e integração com roteamento de veículos**. Tese (Doutorado em Engenharia Elétrica) – Universidade Estadual de Campinas, Campinas S.P., 2006.
- AUTODESK, EUA. **AutoCAD: Reference Manual**. 2010.
- BARROS C. M.P., JUNIOR L. C., ROCHA M. L. **Uma Metaheurística GRASP aplicada ao problema de corte guilhotinado bi-dimensional**. 2004.
- BLUM, C. E ANDREA, R. **Metaheuristics in combinatorial optimization: Overview and conceptual comparison**, ACM Computing Survey, 35 (3), pp. 268-308. 2003
- BURKE, B.; HELLIER, R. S. R.; WHITWELL, G. **Complete and robust no-fit polygon generation for the irregular stock cutting problem**. *European Journal of Operational Research* v.179 p. 27–49, 2007.
- BURKE, B.; KENDALL, G.; WHITWELL, G. **A new placement heuristic for the orthogonal stock cutting problem**. *Oper. Res.* V.52(4) p.655–671, 2004.
- CAVALI, R.; **O Problema de Corte e Empacotamento na indústria de móveis: Um estudo de caso**. Dissertação (Mestrado em Engenharia da Produção), UNESP. São José do Rio Preto. SP. 2004
- CHERRI, A. C. **Algumas extensões do problema de corte de estoque com sobras reaproveitáveis**. Tese (Doutorado em Ciências Matemáticas e da Computação) – Universidade de São Paulo, SÃO CARLOS, 2009.
- CONSTANTINO, A. A.; JUNIOR, G.M.A. **Um algoritmo genético híbrido para o problema de corte industrial bidimensional**. *Acta Scientiarum*. Maringá, v. 24, n. 6, p. 2002.
- COREDEIRO A., MÁSCULO F.S. **Características de uma ferramenta CAD voltada a uma abordagem integrada do processo de projeto e arquitetura**. *Gestão de Produção*. 2005.
- CORTE CERTO. **Software para otimização de planos de corte**. Disponível em <<http://www.cortecerto.com/portugues/downloads.htm>>. 2010.
- COSTA, D. D.; PEREIRA, G. A. **Desenvolvimento e avaliação de uma tecnologia de baixo custo para programação CNC em pequenas empresas**. *Produção*, v. 16, n. 1, p. 048-063, 2006.
- COSTA, M. T. **Novas abordagens ao posicionamento de figuras irregulares**. Tese (Doutorado em Engenharia Elétrica) - Universidade do Porto, Portugal. 2007.
- CUNHA, R. R. **Estudo e desenvolvimento de metodologias na troca de dados em CAD/CAM**. Tese (Doutorado em Engenharia Mecânica)- Universidade Federal de Santa Catarina. 2000.
- DA SILVA, A. D.; BEVILACQUA, J. C. **Implementação de Pós-Processadores em Sistemas CAD/CAM Dedicados**. In: COBEF - Congresso Brasileiro de Engenharia de Fabricação. Joinville. 2005.
- DA SILVA, A. D.; GONÇALVES M. A. F. **Aumente os Recursos de Máquinas CNC**. *Mecatrônica Atual* v. 6, p. 26-29, 2008.
- DA SILVA, A. D.; SOUZA, D. J. A. **Geração de Ciclos Avançados de Usinagem a partir de Ciclos Básicos disponíveis em equipamentos CNC**. In: Congresso Brasileiro de Engenharia de Fabricação – COBEF Uberlândia. 2003.
- DIAZ, A. F. **Optimización Heurística y Redes Neuronales**. Madrid: Paraninfo, 1996.
- DYCKHOFF, H. **A typology of cutting and packing problems**. *European Journal of Operational Research*, 44(2):145–159, 1990.
- FEO, T. A.; RESENDE, M. G. C. **A probabilistic heuristic for a computationally difficult set covering problem**. *Operations Research Letters*, 8(2):67– 71, 1989.
- FERNANDES, A. C. **Proposta de uma metodologia para cortar peles de couro na indústria de calçados**. *Gestão e Produção*. 2000.

- FILHO, P. C. A. **Projeto e Operações de Reatores Auxiliados por Computador**. Dissertação (Mestrado em Engenharia Química) – Universidade Estadual de Campinas. 1999.
- FOGGIATTO, J. A.; VOLPATO, N.; BONTORIN, A.C. **Recomendações para modelagem em sistemas CAD-3d**. Congresso Brasileiro de Engenharia de Fabricação. 2007
- GILMORE, P.; GOMORY, R. **Multistage cutting stock problems of two and more dimensions**. *Operations Research*, **14**, 94-120. 1965.
- GOLFETO, R. R.; MORETI, A. C.; NETO L. S. **A GRASP metaheuristic for the ordered cutting stock problem**. Revista chilena de ingeniería, vol. 16 N° 3. 2008
- GOMES, A. M. **Uma Biblioteca Geométrica e Eficiente para a Resolução de Problemas de Corte de Formas Irregulares**. Disponível em < http://www.fe.up.pt/si/projectos_geral.mostra_projecto>. 2008. Acesso em 10/07/2009.
- GOMES, A. M.; COSTA, M. T.; OLIVEIRA, J. F. **An iterated local search approach to the large-scale regular packing of irregular shapes in a limited sheet**. In MIC 2005 – The 6th Metaheuristics International Conference, p 421–428. Vienna, 2005.
- HOPPER, E. ; TURTON, .B. C.H. **An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem**. European Journal of Operational Research **128**, 2001.
- HOPPER, E. **Two-dimensional packing utilizing evolutionary algorithms and other meta-heuristic methods**. Ph.D. thesis, *University of Wales*, Cardiff, UK. 2003.
- KONG, L.; FUH J. Y. H.; LEE, X. L.; LING, L. S. **A Windows-native 3D plastic injection mold design system**. Journal of Materials Processing Technology. p. 1-9. 2003.
- LEDA. **Library of Efficient Data Types and Algorithm**. Disponível em: <<http://www.algorithmic-solutions.com/enleda.htm>>. 2008.
- LODI, A.; MARTELO, S. E.; VIGO, D. **Models and bounds for two-dimensional level packing problems**. Journal of Combinatorial Optimization, v. 8, p. 363–379. 2004.
- MARTÍNEZ, R. R. **Seqüencia para Selecionar Sistemas de Diseño y Manufactura Asistidos por Computadora (CAD/CAM)**. . Información Tecnológica-Vol. 15 N° 2-2004, págs.: 91-94
- MENEGHELLO, P. G. **Aplicação de um sistema robótico utilizando recursos de sistemas CAD/CAM para o processo de fresamento**. 2003. Dissertação (Mestrado em Engenharia Mecânica) Universidade Federal do Rio Grande do Sul.
- MIRANDA M. **Algoritmos genéticos fundamentos e aplicações**. Disponível em : <<http://www.gta.ufrj.br/~marcio/genetic.html>> 2005. Acesso em 10 julho 2010.
- MONARO, G. R. L.; SCHUTZER K. **Capacitação em modelamento e manufatura de sólidos utilizando sistema CAD/CAM**. Revista De Ciência & Tecnologia, 2005.
- MORABITO **Modelos de otimização para o problema de corte nas indústrias de papel e papelão e de móveis**. Gestão & Produção, v. 1, n. 1, p. 59-76, abr. 1994
- MORABITO, R.; ARENALES, M. **Otimização das Operações de Corte de Chapas na Indústria de Móveis**. In: Oficina Nacional de PCE, XX CNMAC, 2, 1998, Curitiba. Anais. P. 43-52,1998.
- MOSQUERA, P. G. **Redução de ciclos da serra no problema de corte de estoque bidimensional na indústria de móveis**. São José do Rio Preto, São Paulo, 2008.☒
- MÜLLER, F. M. **Algoritmos heurísticos e exatos para resolução do problema de sequenciamento em processadores paralelos**. Tese (Doutorado de Engenharia Elétrica) Universidade Estadual de Campinas. Campinas, 1993.
- Nakamura, T. E.; JUNIOR, A. M. J; ZANOLLI, J.; MACHADO, W. **Utilização de ferramentas CAD/CAE/CAM no desenvolvimento de produtos eletroeletrônicos: vantagens e desafios**. T&C Amazônia, Ano 1, no 2, 2003,
- NEVES, J. A. **Desenvolvimento de um dispositivo eletromecânico para a troca automática de ferramentas em centros de torneamento CNC**. Dissertação (Mestrado em Engenharia Mecânica – Automação Industrial e Robótica), Universidade de Taubaté, Taubaté, 2005.
- NORONHA, F. T. **Uma abordagem sobre estratégias metaheurísticas**. Universidade Federal do Rio Grande do Norte. 2005.
- NORONHA, T. F.;SILVA, M.; ALOISE, J. D. **Uma Abordagem sobre Estratégias Meta-heurísticas**. 2005

- OLIVEIRA J. F., GOMES, A. M. **A new constructive algorithm for nesting problems.** *OR Spektrum*. V. 22 p.263–284, 2005.
- OLIVEIRA, E J. F. **A 2-exchange heuristic for nesting problems.** *European Journal of Operational Research*, v. 141(2) p.359–370, 1995.
- OLIVEIRA, J. F. **Problemas de Posicionamento de Figuras Irregulares: uma perspectiva de otimização.** Tese de Doutorado, Universidade do Porto,1995.
- OLIVEIRA, J. F., FERREIRA, J. S. **Algorithms for nesting problems.** *European Journal of Operational Research*,v. 84 p. 506–521, 1990.
- OSMAN, I. H. **Focused issue on applied meta-heuristics.** *Computers and Industrial Engineering*. Vol. 44, p. 205-207, 2002.
- PILEGGI, C .G.; MORABITO, R.; ARENALES, M. N. **Heurísticas para os problemas de geração e sequenciamento de padrões de corte bidimensionais.** *Pesquisa Operacional*, v.27, n.3, p.549-568, 2007.
- PINTO, M.. **Algumas contribuições à resolução do problema de corte integrado ao problema de sequenciamento dos padrões.** Tese (Doutorado em Computação Aplicada) – Instituto de Pesquisa Espaciais São José, 2004.
- POLDI, K. C., ARENALES, M. N. **Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths.** *Computers and Operations Research*, 36: 2074-2081, 2009.
- POLL, M. **Uma metodologia para automação do processo de conformação por calandras.** Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de Santa Maria. 2005,
- Prais, M. **Parameter variation in GRASP implementations.** In Proc. of the Third Metaheuristics International Conference (1999)
- PRAIS, M.; RIBEIRO, C.C. **Parameter variation in GRASP implementations.** In Proc. of the Third Metaheuristics International Conference, pg 07 e 08. Angra dos Reis (RJ). 1999.
- RANGEL, S.; FIQUEIREDO, A. **O problema de corte de estoque em indústrias de móveis de pequeno e médio portes.** *Pesquisa Operacional*, v.28, n.3, p.451-472, 2008.
- REEVES, C. R. **Modern Heuristic Techniques for Combinatorial Problems.** New York: John Wiley & Sons, Inc, 1996, 320p.
- RESENDE, M. G. C. **Greedy Randomized Adaptive Search Procedures (GRASP).** Relatório Técnico Technical Report: 98.41.1, AT&T Labs Research,1998
- RESENDE, M. G. C.; RIBEIRO, C. C. **Greedy randomized adaptive search procedures.** In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pp. 219–249, Kluwer Academic Publishers, 2003.
- SCHMIDT, A. M.; KOHMANN, H.; KUNRATH A. **Fabricação de implantes bucomaxilofaciais em titânio.** 2002.
- SILVA, A. D. da; CARPES, F. P.; BITTENCOURT, W. S.; MOTA, C. B. **Processamento de Imagens aplicado a Programação CNC.** *Revista Máquinas e Metais*, São Paulo - SP, v. 482, n. Março, p. 170-177, 2006.
- SIMÕES, F. M. S. **Implementação de um Sistema CAD/CAM para Fresadora CNC a partir de Funções CAM Integradas no CAD.** Uberlândia, MG. Dissertação de mestrado. Universidade Federal de Uberlândia - UFU, 110 p. 1999.
- SOUZA, F.A.; COELHO, T.G. **Tecnologia CAD/CAM - Definições e estado da arte visando auxiliar sua implantação em um ambiente fabril.** XXIII Encontro Nac. de Eng. de Produção - Ouro Preto, MG, 2003.
- SPECK, H. J. **Proposta de método para facilitar a mudança de técnicas de projetos: da prancheta à modelagem sólida (CAD) para empresas de engenharia de pequeno e médio porte.** 2005. 185 f. Tese (Doutorado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis.
- SULIMAN A. M. S. **A sequential heuristic procedure for the two-dimensional cutting stock problem.** *Int. J. Production Economics* 99 (2006) 177–185. 2005.
- TAKASE K. F.; TSUZUKI, G.S.M.;VOGEL, N.; **Modelador de Sólidos Didático.** SAE. 2005. . In: Congresso Brasileiro de Engenharia Mecânica, 15., Águas de Lindóia,2005.
- TEBIS, S. **A evolução do planejamento e controle da produção.** Disponível em <www.abepro.org.br/biblioteca/ENEGEP2006_ART265.pdf> . 2006.

-
- TEMPONI, C. C. E; SOUZA, R. S.; ANDRADE, M. S.; SANTOS, F. A. **Open Dimensional Cutting Problem**: uma abordagem Híbrida via GRASP e ILS. VII Encontro de Engenharia de Produção. 2007.
- VELASCO, S. A. **GRASP para o problema de corte bidimensional guilhotinado e restrito**. Dissertação (Mestrado em Ciências de Engenharia) - Universidade Estadual do Norte Fluminense, 2008.
- VELASCO, S. A.; JUNIOR, P.P; VIEIRA NETO, E. **Um algoritmo heurístico baseado na GRASP para o problema de corte bidimensional guilhotinado**. GEPROS – Gestão da Produção. 2009.
- VIEIRA NETO, E. **GRASP: efeito da independência das soluções iniciais na otimização de corte**. Tese (Doutorado em Ciências da Engenharia) – Universidade Estadual do Norte Fluminense. CAMPOS DOS GOYTACAZES. 2004.
- WASCHER, G.; HAU_NER, H.; SCHUMANN, H. **An improved typology cutting and packing problems**. European Journal of Operational Research, 183:1109-1130. 2007.
- YANASSE, H.H.; CERQUEIRA, L. G. **Uma heurística baseada em geração sequencial de padrões para o problema de corte de estoque unidimensional com número reduzido de padrões**. Gestão da Produção, São Carlos, v. 16, n. 2. 2009.
- ZEID, I. **Mastering CAD/CAM**. New York: McGraw Hill, 2005, 962p.