

**GRASP PARA O PROBLEMA DE CORTE BIDIMENSIONAL
GUILHOTINADO E RESTRITO**

ANDRÉ SOARES VELASCO

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE – UENF

CAMPOS DOS GOYTACAZES – RJ
AGOSTO – 2005

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

**GRASP PARA O PROBLEMA DE CORTE BIDIMENSIONAL
GUILHOTINADO E RESTRITO**

ANDRÉ SOARES VELASCO

Dissertação submetida ao corpo docente do Centro de Ciência e Tecnologia da Universidade Estadual do Norte Fluminense, como parte das exigências necessárias para obtenção do grau de Mestre em Ciências de Engenharia, na área de concentração de Engenharia de Produção.

Orientador: Prof. Geraldo Galdino de Paula Junior, D.Sc.

CAMPOS DOS GOYTACAZES – RJ
AGOSTO – 2005

GRASP PARA O PROBLEMA DE CORTE BIDIMENSIONAL
GUILHOTINADO E RESTRITO

ANDRÉ SOARES VELASCO

Dissertação submetida ao corpo docente do Centro de Ciência e Tecnologia da Universidade Estadual do Norte Fluminense, como parte das exigências necessárias para obtenção do grau de Mestre em Ciências de Engenharia, na área de concentração de Engenharia de Produção.

Aprovada em 17 de agosto de 2005

Comissão Examinadora:

Prof. José Ramón Arica Chavéz, D.Sc. - UENF

Prof. José Elias Cláudio Arroyo, D.Sc - UCAM-Campos

Prof. Euclides Vieira Neto, D.Sc. – Estácio de Sá

Prof. Geraldo Galdino de Paula Junior, D.Sc. - UENF
Orientador

DEDICATÓRIA

A minha carinhosa mãe (†) Hudinéa,
ao meu corajoso irmão (†) Julinho e aos
meus queridos avós (†), Délphica e Humberto.

AGRADECIMENTOS

Agradeço a Deus, por ter me conduzido e iluminado em todos os momentos.

Ao meu orientador Geraldo Galdino de Paula Jr., pela valiosa orientação, amizade, compreensão e, sobretudo, pela confiança depositada no desenvolvimento deste trabalho.

Ao professor Euclides Vieira Neto e família, pelo apoio, amizade e carinho imprescindíveis para essa conquista. A vocês minha eterna gratidão.

Aos professores José Ramón Arica Chavéz e José Elias Cláudio Arroyo, membros da comissão examinadora, e ao professor Rodrigo Nogueira pelas sugestões apresentadas para o aperfeiçoamento deste trabalho.

A FAPERJ pelo apoio financeiro durante o período de mestrado.

Aos meus amigos, colegas de mestrado e funcionários da UENF: Frederico Galaxe, Leonardo Póvoa, Rodrigo Resende, Alander Ornelas, Eduardo Varejão, Carlos Vargas, Érica Azevedo, Ailton Ferreira, Érik Oliveira, Luis Carlos de Oliveira, Edson Braga, Rildo Mororó, Regina Franco, Augusto Pimentel, Edílson Maciel, Rogério de Castro, Vânia Navarro, Gilberto Jubini, Luis Henrique Jassus, e tantos outros que, direta ou indiretamente, contribuíram para realização deste trabalho, não hesitando momento algum em me incentivar.

Ao meu pai, Julio da Silva Velasco, e a todos da minha família pelo apoio irrestrito e palavras de encorajamento nos momentos difíceis.

Ao meu filho Pedro Julio pela sua presença, carinho e compreensão durante a realização deste trabalho.

“Deus é o grande geômetra.
Deus geometriza sem cessar”.
(Platão)

SUMÁRIO

Lista de Figuras	VIII
Lista de Tabelas	X
Lista de Quadros	XI
Resumo	XII
Abstract	XIII
Capítulo 1 – Introdução	1
Capítulo 2 – Tipologia dos Problemas de Corte e Empacotamento	6
2.1 - Dimensionalidade	8
2.2 - Medidas Quantitativas	10
2.3 - Formato das Figuras	10
2.4 - Sortimento	11
2.5 - Disponibilidade	11
2.6 - Restrições de Padrão	12
2.7 - Restrições de Alocação	14
2.8 - Objetivos	15
2.9 - Estado da Informação e Variabilidade	16
2.10 - Nomenclatura de Dyckhoff	16
Capítulo 3 – O Problema de Corte Bidimensional	18
3.1 - O Problema de Corte Bidimensional Guilhotinado e Restrito	20
3.2 - Formulação do Problema de Corte Bidimensional Guilhotinado e Restrito	22
3.3 - Soluções Aproximadas	28
3.4 - Revisão Bibliográfica	29

Capítulo 4 – Métodos Heurísticos	35
4.1 - <i>Simulated Annealing</i>	37
4.2 - Busca Tabu	38
4.3 - GRASP	39
4.3.1 - Fase de Construção da GRASP	40
4.4.2 - Fase de Melhoria da GRASP	42
Capítulo 5 – Implementação do algoritmo GRASP	44
5.1 - Aplicação da GRASP ao Problema de Corte Bidimensional	48
5.1.1 - Fase de Construção do Algoritmo GRASP-2D	48
5.1.1 - Fase de Melhoria do Algoritmo GRASP-2D	49
5.2 - Algoritmo GRASP-2D	50
Capítulo 6 – Resultados Computacionais	56
6.1 - Primeira Classe de Instâncias	57
6.2 - Segunda Classe de Instâncias	66
Capítulo 7 – Conclusões	87
REFERÊNCIAS BIBLIOGRÁFICAS	89
APÊNDICE	94
ANEXO 1: Instâncias Disponíveis na Literatura	97
ANEXO 2: Instâncias Geradas Aleatoriamente	100

LISTA DE FIGURAS

Figura 2.1	- Estrutura dos problemas de corte e empacotamento	7
Figura 2.2	- Padrão de corte para o problema bidimensional	8
Figura 2.3	- Padrão de corte para o problema tridimensional	9
Figura 2.4	- Padrão ortogonal bidimensional guilhotinado	13
Figura 2.5	- Padrão ortogonal bidimensional não guilhotinado	14
Figura 3.1	- Peças regulares e irregulares	18
Figura 3.2	- Corte guilhotinado ortogonal	19
Figura 3.3	- Cortes guilhotinados em dois estágios	20
Figura 3.4	- Peças a serem cortadas	21
Figura 3.5	- Peças a serem produzidas	21
Figura 3.6	- Padrões de corte	22
Figura 5.1	- Cortes guilhotinados e faixas guilhotina	44
Figura 5.2	- Faixas guilhotina horizontais e verticais	46
Figura 5.3	- Perda Externa e Perda Interna	47
Figura 6.1	- Comparativo dos resultados executando os algoritmos para as instâncias da primeira classe	59
Figura 6.2	- Padrão de corte produzido para L1	60
Figura 6.3	- Padrão de corte produzido para L2	61
Figura 6.4	- Padrão de corte produzido para L3	62
Figura 6.5	- Padrão de corte produzido para L4	63
Figura 6.6	- Padrão de corte produzido para L5	64
Figura 6.7	- Padrão de corte produzido para L6	65
Figura 6.8	- Comparativo dos resultados executando os algoritmos para as instâncias da segunda classe	68
Figura 6.9	- Padrão de corte produzido para A1	69
Figura 6.10	- Padrão de corte produzido para A2	70
Figura 6.11	- Padrão de corte produzido para A3	71
Figura 6.12	- Padrão de corte produzido para A4	72
Figura 6.13	- Padrão de corte produzido para A5	73
Figura 6.14	- Padrão de corte produzido para A6	74
Figura 6.15	- Padrão de corte produzido para A7	75

Figura 6.16 - Padrão de corte produzido para A8	76
Figura 6.17 - Padrão de corte produzido para A9	77
Figura 6.18 - Padrão de corte produzido para A10	78
Figura 6.19 - Padrão de corte produzido para A11	79
Figura 6.20 - Padrão de corte produzido para A12	80
Figura 6.21 - Padrão de corte produzido para A13	81
Figura 6.22 - Padrão de corte produzido para A14	82
Figura 6.23 - Padrão de corte produzido para A15	83
Figura 6.24 - Padrão de corte produzido para A16	84
Figura 6.25 - Padrão de corte produzido para A17	85
Figura 6.26 - Padrão de corte produzido para A18	86

LISTA DE TABELAS

Tabela 6.1 - Resultados da primeira classe de instâncias com o GRASP-2D	58
Tabela 6.2 - Resultados obtidos da primeira classe de instâncias	58
Tabela 6.3 - Resultados da segunda classe de instâncias com o GRASP-2D	66
Tabela 6.4 - Resultados obtidos da segunda classe de instâncias	67

LISTA DE QUADROS

Quadro 2.1 - Nomenclatura da Tipologia	16
Quadro 2.2 - Tipologia de alguns problemas	17
Quadro 6.1 - Referências bibliográficas da primeira classe de instâncias	57

Resumo da dissertação apresentada ao CCT/UENF como parte dos requisitos necessários à obtenção do grau de Mestre em Ciências de Engenharia (M.Sc.), na área de Engenharia de Produção.

GRASP PARA O PROBLEMA DE CORTE BIDIMENSIONAL GUILHOTINADO E RESTRITO

André Soares Velasco

17 de Agosto de 2005

Orientador: Prof. Geraldo Galdino de Paula Junior, D.Sc.

Esta dissertação de mestrado tem como objeto de estudo um caso particular dos problemas de corte, conhecido na literatura como problema de corte bidimensional guilhotinado e restrito. Este problema consiste em determinar a melhor forma de se produzir uma quantidade de peças retangulares, realizando cortes do tipo guilhotina, em placas retangulares em estoque. Considerando a sua complexidade combinatorial e a diversidade de aplicações práticas, este problema é abordado via métodos heurísticos. Dentre as metodologias usadas com relativo sucesso para o tratamento deste problema sobressaem-se às heurísticas Busca Tabu e *Simulated Annealing*. Neste trabalho é apresentada uma formulação heurística, fundamentada na metodologia GRASP, como proposta para resolução do problema de corte enfatizado. Com o intuito de validar a heurística proposta, foram realizados testes computacionais em instâncias encontradas na literatura. Os resultados apresentados foram bastantes significativos comparados às soluções encontradas com as heurísticas Busca Tabu e *Simulated Annealing*.

Palavras Chave: problema de corte, padrão de corte bidimensional guilhotinado e restrito, heurística, GRASP.

Dissertation abstract presented to CCT/UENF as part of the needed demands to get the Master degree in Engineering Sciences (M.Sc.), in the area of Production Engineering.

GRASP FOR CONSTRAINED AND GUILLOTINE TWO-DIMENSIONAL CUTTING PROBLEM

André Soares Velasco

August 17th, 2005

Advisor: Prof. Geraldo Galdino de Paula Junior, D.Sc.

This master's degree dissertation has the objective to study a particular case of cutting problems, known in the literature as constrained and guillotine two-dimensional cutting problems. This problem consist in determining the best form to produce a quantity of rectangles pieces, making cuts like guillotine kinds, in rectangles plates in store. Considering its combinatorial complexity and a variety of practical application, this problem is approached by heuristic method. Among the methodology used with relative success for treatment in this case, stands out the Tabu Search and Simulated Annealing heuristics. In this work is presented a heuristic formulation based on the GRASP methodology, as proposition to solve the cutting problems. In order to validate the heuristic propose it was realized computer tests in instances found in the literature. The presented results were very meaningful, compared to solutions found in the Tabu Search and Simulated Annealing heuristics.

Keywords: cutting problem, two-dimensional constrained guillotine cutting patterns, heuristic, GRASP.

1. INTRODUÇÃO

As empresas que usam o corte de matéria-prima, com a finalidade de produzir produtos semi-acabados ou finais, têm dado grande atenção ao problema das perdas de material no processo produtivo, indo em busca de soluções que reduzam o desperdício do material durante o processo de corte. Logo, a otimização do corte nestas empresas é de grande relevância para o planejamento da produção.

A forma pela qual é processado o corte em metais, vidros, rochas ornamentais, borrachas, tecidos e outros materiais, geralmente, melhora com a experiência adquirida pelo setor produtivo responsável pelo corte, mas o surgimento de novas tecnologias de corte estabelece novas condições às empresas para se manterem no cenário que atuam. Entretanto, nem todas as empresas têm acesso a essas tecnologias, principalmente as pequenas e médias empresas, em decorrência do alto investimento na aquisição destes equipamentos ou pela falta de informação por parte da administração.

As empresas bem sucedidas consideram imprescindível um rigoroso controle de seus custos. Com a alta competitividade, uma empresa deve estar preparada para sobreviver e prosperar no mercado. Realizar as atividades de forma apropriada e com o menor custo possível proporcionam a uma empresa considerável vantagem competitiva.

Todas as operações produtivas utilizam algum tipo de tecnologia de processo e devem ser continuamente melhoradas. Slack et al (1996, p. 252) considera que “a tecnologia de processos constitui-se de máquinas, equipamentos e dispositivos que ajudam a produção a transformar materiais de forma a agregar valor e atingir os objetivos estratégicos da produção”.

A produtividade nos dias atuais se faz importante, a qualidade gera a produtividade, utilizando, de forma inteligente, os recursos da produção. As empresas que desejam aumentar sua competitividade no mercado devem ter como objetivos a alta qualidade e a eficiência, pois fazer de forma errônea pode significar

altos custos. Diminuir o desperdício de matéria-prima no processo produtivo reduz os custos e, conseqüentemente, esses produtos podem chegar ao mercado com um preço menor, aumentando o volume de vendas, a receita da empresa e a satisfação do consumidor. Os custos de uma operação podem ser reduzidos melhorando a eficiência do processo.

Vários problemas de engenharia podem ser representados, através de um modelo matemático, como problemas de otimização. Um desses é conhecido como Problema de Corte (*Cutting Stock Problem*). O problema de corte origina-se na produção de um conjunto de peças feitas a partir de uma peça maior, de modo que a perda durante o processo de corte seja minimizada. Na maioria das vezes, a forma como se determina o arranjo dos cortes não é fundamentada em métodos científicos, e sim por métodos, que, baseados apenas na experiência, acabam produzindo sobras de material em quantidades indesejáveis. Em alguns casos essas sobras não podem ser aproveitadas na produção de peças úteis, refletindo diretamente no custo final do produto.

Esta dissertação considera a otimização do corte guilhotinado de peças retangulares, com demandas definidas e dimensões que podem diferir, a partir de uma placa maior, também com formato retangular, existente em estoque. Citado na literatura como Problema de Corte Bidimensional Guilhotinado e Restrito, este é bastante relevante em diversos setores produtivos, tais como: metal-mecânica, moveleira, vidraceiro, beneficiamento de rochas ornamentais, entre outros. O problema é bidimensional devido à importância da largura e do comprimento nas chapas a serem cortadas e nas peças obtidas. O corte é dito guilhotinado se, aplicado a uma placa retangular, produzir duas novas placas retangulares. Seja um padrão de corte determinado por um possível arranjo de peças em uma placa, este é dito restrito se há uma limitação na quantidade de peças obtidas, ou seja, o número de determinada peça a ser obtida com os cortes efetuados nas placas deve satisfazer a sua demanda.

Do ponto de vista prático, o problema de corte bidimensional abordado nesta dissertação está presente no dia-a-dia das várias empresas do setor de beneficiamento de rochas ornamentais encontradas nas regiões Norte e Noroeste

Fluminense. As marmorarias, como são conhecidas essas empresas, vendem peças sob encomenda, que são produzidas cortando-se as chapas de mármore e granito em pedaços menores. O arranjo dos cortes numa chapa pode ser de várias maneiras e, geralmente, produzem retalhos de matéria-prima, provenientes dos pedaços que sobram, após o corte das peças desejadas. Nos casos em que a otimização das chapas não se faz presente, os retalhos, sobras de matéria-prima que não são reaproveitadas, acabam influenciando o preço do produto final, devido ao valor irrelevante destes retalhos no mercado. Minimizar o desperdício de matéria-prima, com a melhoria e a eficiência das atividades de corte, em empresas de beneficiamento de rochas ornamentais, pode resultar em economias substanciais vindo a ser um importante diferencial de competitividade para as empresas do setor.

A fácil compreensão do problema de corte bidimensional esconde a sua real complexidade quando o objetivo é obter soluções rápidas, que atendam as necessidades cotidianas dos setores produtivos em tempo real.

Os problemas de cortes de peças pertencem à classe de problemas denominada *NP-Hard* (GAREY; JOHSON, 1979), em que não é interessante o uso de algoritmos exatos para problemas de médio e grande porte. Esta inviabilidade é atribuída à explosão combinatória de arranjos possíveis, quando o objetivo é a determinação de um arranjo ótimo. Isto significa que os tempos computacionais para obtenção de soluções ótimas, através de métodos exatos, são grandes e com a tecnologia atual não se pode contar com resultados práticos. Nestes casos, os métodos heurísticos são bastante considerados, constituindo-se uma alternativa válida para o problema.

A metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) é uma técnica iterativa composta de duas fases: uma fase de construção, na qual soluções iniciais são geradas, e uma fase típica de melhoria. Os resultados, bastante satisfatórios, apresentados por Vieira Neto (1999; 2004), utilizando a GRASP em problemas de corte unidimensional, foram determinantes para o uso desta metodologia na solução dos problemas de corte bidimensional.

Este trabalho tem como objetivo apresentar um método de solução para problema de corte bidimensional e restrito, com cortes guilhotinados, utilizando a técnica heurística GRASP. Os resultados computacionais obtidos com a utilização do algoritmo proposto GRASP-2D, e os resultados encontrados em testes já realizados, com o Busca Tabu e o *Simulated Annealing* (AMARAL, 1994), são comparados quando aplicadas a determinadas instâncias do problema.

A dissertação está dividida em sete capítulos. Além do Capítulo 1 introdutório, uma estrutura lógica, compreendendo as características essenciais para a classificação dos Problemas de Corte e Empacotamento, segundo Dyckhoff (1990), será descrita no Capítulo 2.

No Capítulo 3, serão mostradas algumas definições do Problema de Corte Bidimensional relevantes ao problema objeto desta dissertação. Ainda neste capítulo, serão apresentadas as dificuldades envolvidas na modelagem do problema de corte bidimensional restrito, com cortes guilhotinados, finalizando com uma revisão bibliográfica.

O Capítulo 4 será destinado à apresentação dos métodos heurísticos, utilizados pela maioria dos autores no tratamento dos problemas de corte bidimensional. No desenvolvimento deste capítulo, será feita uma descrição das metaheurísticas envolvidas neste trabalho, sendo a GRASP abordada com destaque.

No Capítulo 5 será apresentada a implementação computacional do algoritmo GRASP-2D proposto. Este capítulo iniciará-se com as definições necessárias para a arquitetura do algoritmo. Em seguida, serão apresentados os procedimentos característicos da metodologia GRASP, fase de construção e fase de melhoria, desenvolvidos para otimização do problema de corte bidimensional guilhotinado e restrito. No final deste capítulo, será descrito o pseudocódigo do algoritmo GRASP-2D.

No Capítulo 6 serão apresentados os resultados, referentes ao padrão de corte com maior percentual de aproveitamento da placa, obtidos nos testes

computacionais com o algoritmo proposto. A partir destes resultados, o desempenho do algoritmo GRASP-2D será comparado com o desempenho dos algoritmos Busca Tabu e *Simulated Annealing* (AMARAL, 1994).

Finalmente, no Capítulo 7, serão apresentadas algumas conclusões e sugestões para novas pesquisas.

2. TIPOLOGIA DOS PROBLEMAS DE CORTE E EMPACOTAMENTO

O Problema de Corte consiste, essencialmente, em determinar a melhor maneira de produzir um conjunto de peças menores (denominadas itens) efetuando cortes em peças maiores (denominadas objetos), enquanto, o Problema de Empacotamento consiste, basicamente, em determinar o melhor arranjo de um conjunto de itens dentro de objetos. Empacotar itens dentro de objetos também pode ser visto como cortes de espaços, sendo estes espaços ocupados por itens ou considerados perdas. De forma recíproca, o problema de corte pode ser encarado como empacotamento de espaços ocupados por itens em espaços ocupados por objetos (DYCKHOFF, 1990).

Os problemas de corte e empacotamento, tais como corte de barras de ferro e barrotes de madeira na construção civil, corte de placas de granito nas marmorarias, corte de espumas para colchões e empacotamento de caixas em contêineres estão centrados em objetos e itens definidos por uma, duas ou três dimensões do espaço Euclidiano. De forma análoga, problemas desta categoria podem ter objetos e itens com dimensões abstratas, como por exemplo, o balanceamento de uma linha de montagem, com dimensões temporais. Neste caso, o estoque de objetos é definido pelas estações de trabalho, com intervalos de tempo fixados, e a lista de itens é dada por tarefas específicas com durações que devem ser determinadas. Exceto pela restrição que impõe um ordenamento nas tarefas a cumprir, a estrutura lógica é a mesma encontrada nos problemas de empacotamento.

A figura 2.1 exibirá uma estrutura dos problemas de corte e empacotamento, evidenciando a questão das dimensões espaciais e abstratas. E também, nos exemplos citados serão apresentadas algumas das aplicações para os problemas dessa natureza.

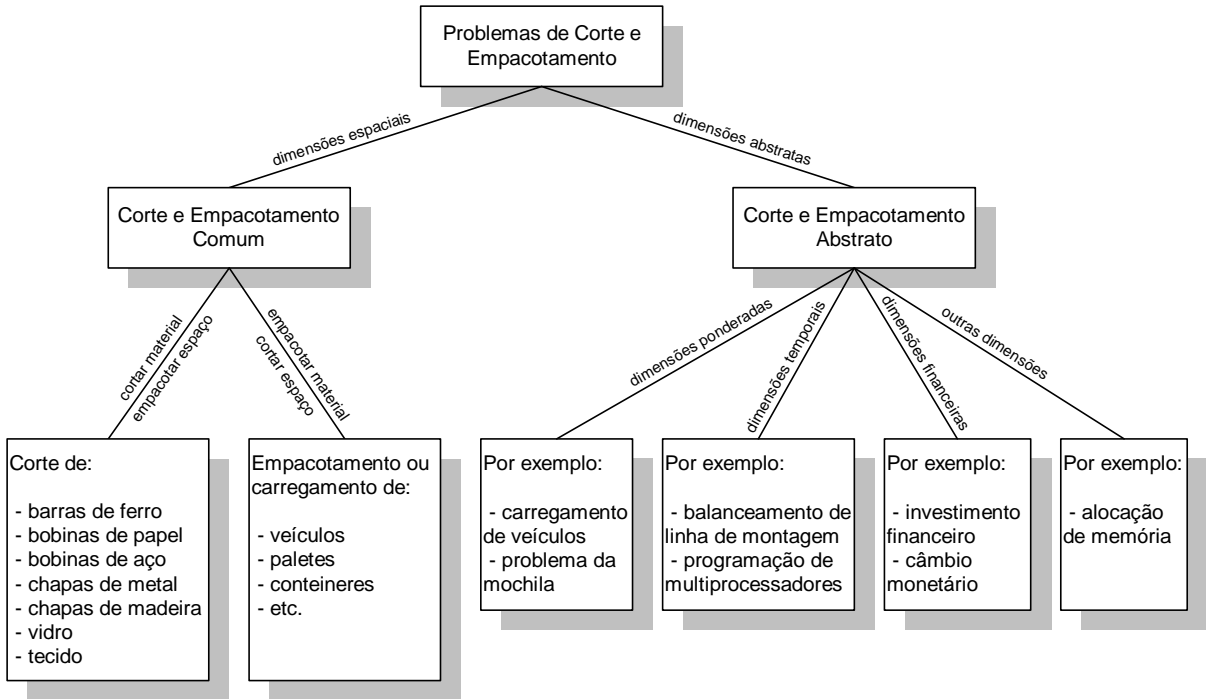


Figura 2.1 - Estrutura dos problemas de corte e empacotamento.

Em vista da multiplicidade dos problemas de corte e empacotamento nos setores produtivos e da importância destes problemas no planejamento da produção de algumas empresas há um incessante interesse de administradores, economistas, engenheiros, matemáticos, entre outros pesquisadores, em buscar soluções para estes problemas. Com a finalidade de classificar os problemas de corte e empacotamento, identificar características comuns e concentrar pesquisas futuras em problemas de tipos específicos, Dyckhoff (1990) sugeriu em seu artigo uma nomenclatura baseada em uma estrutura lógica. A notoriedade deste trabalho culmina na publicação do livro *Cutting and Packing in Production and Distribution*, por Dyckhoff e Finke (1992), relacionando esta estrutura lógica junto à realidade dos problemas de corte e empacotamento.

Esta estrutura lógica inclui características suficientes para especificar o tipo de um problema de corte e empacotamento. A seguir, uma precisa descrição destas características é apresentada:

2.1. DIMENSIONALIDADE

A dimensionalidade é a característica mais importante, já que determina o número de dimensões relevantes, em objetos e itens, na definição de um padrão para o problema. Um padrão corresponde a uma disposição de itens em objetos. Quanto à dimensionalidade, um problema pode ser classificado como:

- Unidimensional;
- Bidimensional;
- Tridimensional;
- Multidimensional;
- $n\frac{1}{2}$ - dimensional.

O problema é unidimensional quando uma única dimensão é relevante para a definição de um padrão. Por exemplo, na indústria de papel, bobinas de comprimento padronizado devem ser cortadas em rolos de vários comprimentos pré-determinados. Mencionado anteriormente na página 6, o balanceamento de uma linha de montagem é um problema unidimensional.

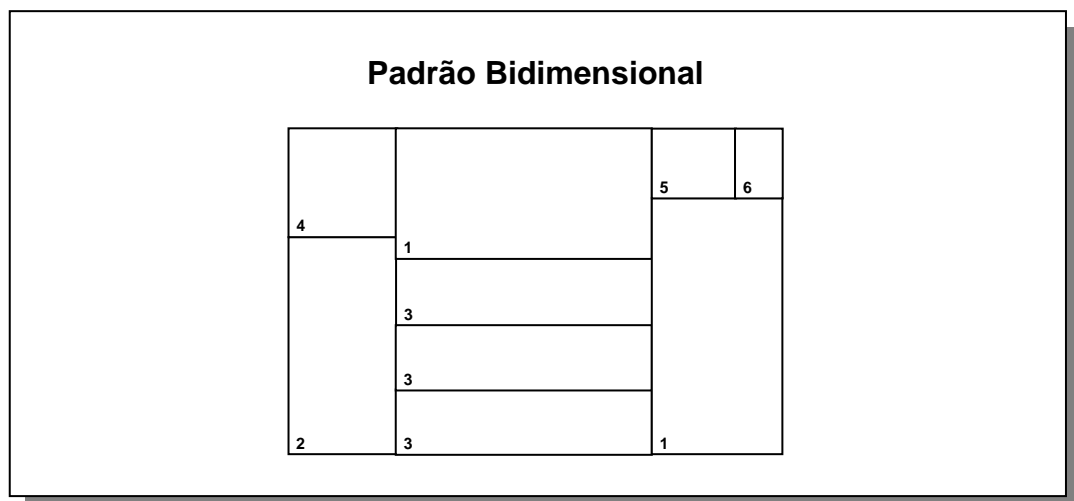


Figura 2.2 - Padrão de corte para o problema bidimensional.

Quando duas dimensões são relevantes na determinação de um padrão, o problema é bidimensional. No corte de chapas de granito em peças menores, nas empresas do setor de beneficiamento de rochas ornamentais (marmorarias), a largura e o comprimento são as medidas relevantes. A figura 2.2 ilustra um padrão de corte para o problema bidimensional.

No corte de colchões a partir de blocos de espumas ou no empacotamento de caixas em contêineres, a largura, o comprimento e a altura, dos objetos e itens, são imprescindíveis na resolução do problema. Este problema é dito tridimensional devido às três dimensões relevantes na definição de um padrão. A figura 2.3 exibe um padrão de corte para o problema tridimensional.

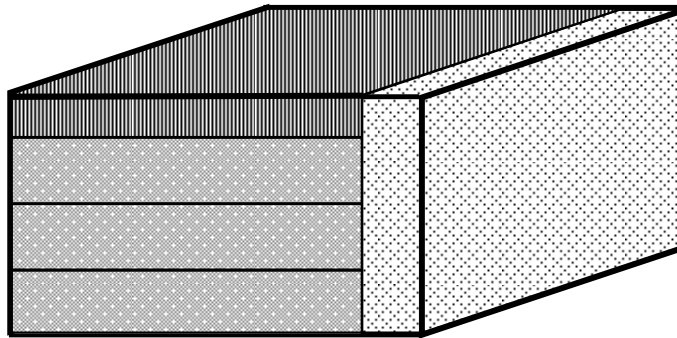


Figura 2.3 - Padrão de corte para o problema tridimensional.

Se o número de dimensões relevantes para definição de um padrão, for maior que três, então este problema é multidimensional. Um exemplo de problema de quatro dimensões seria estocar caixas num contêiner por períodos de tempo fixados e ininterruptos. Neste caso, junto ao problema de empacotamento tridimensional, apresenta-se uma quarta dimensão, o tempo de permanência de cada caixa.

Os problemas $n\frac{1}{2}$ - dimensionais têm $n+1$ dimensões relevantes na definição de um padrão, sendo n dimensões fixas e uma variável. O caso do corte de tecidos, em empresas de publicidade, com intuito de produzir faixas de dimensões

fixas, é um exemplo de problema 1,5 dimensional. Um padrão de corte para este problema, que apresenta 1+1 dimensões relevantes, é definido em um tecido de largura fixa e comprimento variável, ou seja, com comprimento suficiente para atender a demanda.

2.2. MEDIDAS QUANTITATIVAS

Os valores atribuídos às variáveis que indicam o número de objetos e itens em uma solução podem ser:

- Discretos;
- Contínuos.

Em problemas com dimensões bem definidas, como o caso bidimensional, as variáveis estão condicionadas a assumir valores discretos (números naturais). Enquanto nos problemas com uma dimensão variável, há variáveis assumindo valores contínuos (números reais).

2.3. FORMATO DAS FIGURAS

Outra característica a ser observada é o formato das figuras de objetos e itens envolvidos no problema. Seja uma figura distinguida por:

- Forma;
- Tamanho;
- Orientação.

Figuras de mesma forma podem diferir em tamanho ou orientação no espaço relevante. Como por exemplo, no corte de vergalhões para armadura na construção civil (VIEIRA NETO, 1999), onde objetos e itens apresentam comprimentos variados, ou no corte de peças retangulares de vidro, onde é permitida uma rotação de 90 graus em objetos e itens. Em problemas com mais

dimensões relevantes às figuras possuem formas regulares, especialmente retangulares ou em blocos, ou irregulares, como as não-convexas e não-simétricas, típicas nas indústrias têxteis e de calçados (ALVARENGA; GOMES; PARADA, 1994).

O tamanho de uma figura pode ser determinado pela medida de seu comprimento, área ou volume. Este consiste em um importante aspecto, pois o tamanho dos itens em relação aos objetos pode impor dificuldades na resolução de um problema específico. Em um problema bidimensional, o tamanho das figuras pode ser definido pelas suas respectivas áreas.

De acordo com o problema, a posição ou orientação de itens em relação a objetos pode ser fixa, admitindo-se apenas 90 graus de rotação ou permitindo-se qualquer orientação. Nesse estudo, a hipótese de um item ser rotacionado em 90 graus é aceita, com as respectivas figuras sendo consideradas idênticas. Assim, uma peça retangular de comprimento x e largura y não difere de outra peça que possui comprimento y e largura x .

2.4. SORTIMENTO

Tanto o formato quanto a diversidade das figuras de objetos e itens são fundamentais na caracterização de um problema. Por exemplo, a indústria de circuito impresso convive com o problema de cortar chapas retangulares (objetos com formatos idênticos) de fibra de vidro para fabricação de placas (muitos itens com formatos distintos) de circuito impresso (SILVEIRA, 1999).

2.5. DISPONIBILIDADE

Com respeito à disponibilidade de objetos e itens são considerados três fatores:

- Limites, superior e inferior, em sua quantidade;
- Seqüência ou ordem;
- Data de utilização.

De acordo com o problema, a quantidade de objetos e itens pode ser restrita ou irrestrita, isto é, limitada ou ilimitada, respectivamente. Em alguns problemas, os padrões são determinados em uma quantidade limitada de objetos, com itens em quantidades ilimitadas. Já em outros, deseja-se obter padrões que produzam itens em quantidades limitadas ou não, a partir de uma indefinida quantidade de objetos. Por exemplo, no cotidiano das empresas do setor vidraceiro, uma quantidade suficiente de chapas é utilizada na produção de uma determinada quantidade de peças.

Nas metalúrgicas, a necessidade de se respeitar certa ordem para objetos e itens, quanto ao tempo na definição de um padrão, é observado quando barras incandescentes cobertas de aço são produzidas, em um processo seqüencial e sem retardos.

2.6. RESTRIÇÕES DE PADRÃO

Basicamente, as restrições de padrão estão ligadas às características geométricas e operacionais do problema. Estas restrições são esclarecidas e distinguidas em quatro importantes grupos descritos a seguir:

Os espaços entre os itens em um padrão são extremamente importantes em alguns casos, como por exemplo, no corte de chapas de vidro onde estes espaços resultam em perda de matéria-prima. Outra situação relevante em processos produtivos está relacionada ao desperdício de um objeto (matéria-prima) ocasionado por um instrumento cortante. Neste caso, geralmente, esta espessura está adicionada às dimensões dos itens encontrados no padrão em questão.

A posição dos itens, em relação aos mesmos, ou em relação ao objeto, tem de ser levada em consideração, como no caso do carregamento de produtos frágeis.

Podem existir restrições quanto ao número de figuras ou itens em um padrão, como por exemplo, no empacotamento de bombons de chocolate em caixas para comercialização, onde a quantidade de itens é limitada.

O tipo de corte executado e o número de cortes permitidos são essenciais nos problemas em que objetos e itens são retangulares ou em forma de blocos. Em um problema bidimensional, cujo padrão é definido através de cortes guilhotinados e ortogonais, a sua complexidade depende do número de mudanças nas direções de corte (estágios) como também do número de cortes paralelos por estágio. A figura 2.4 apresentará um padrão ortogonal bidimensional guilhotinado com três estágios, quatro itens e uma parte restante.

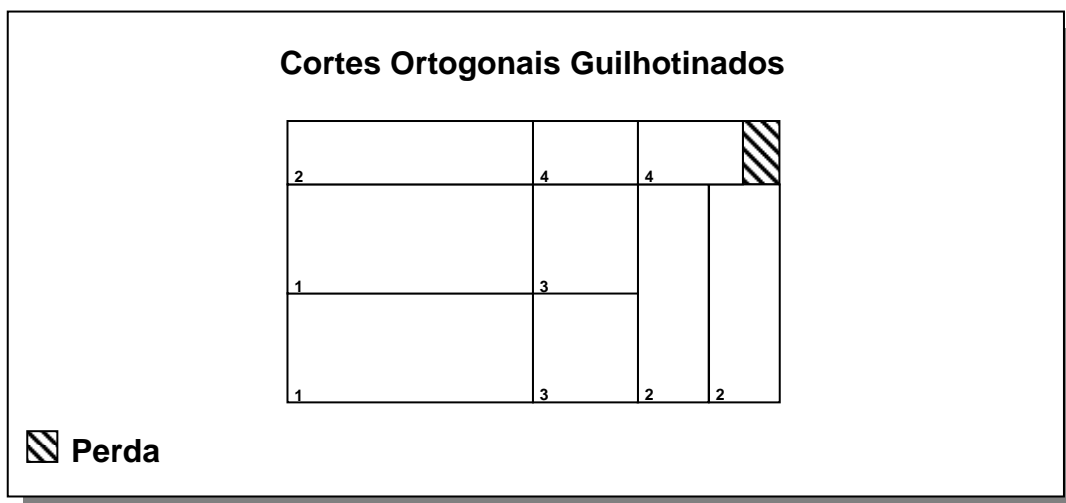


Figura 2.4 - Padrão ortogonal bidimensional guilhotinado.

Pode haver padrões resultantes de cortes não ortogonais e guilhotinados, como também, de cortes ortogonais e não guilhotinados. A figura 2.5 ilustrará um padrão ortogonal bidimensional não guilhotinado com três figuras distintas (dois itens e um objeto) e duas formas diferentes (retangular e quadrada).

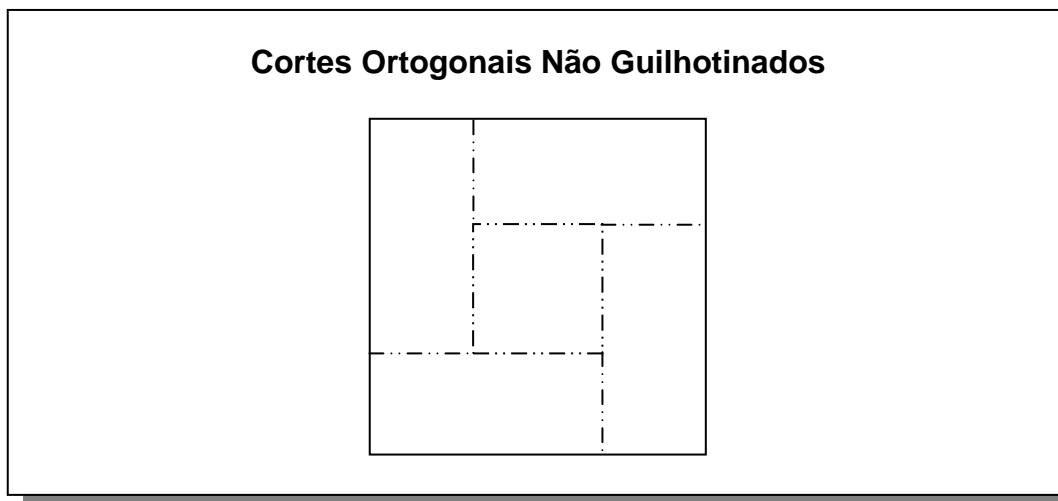


Figura 2.5 - Padrão ortogonal bidimensional não guilhotinado.

2.7. RESTRIÇÕES DE ALOCAÇÃO

Em relação à designação de itens para objetos, são evidenciadas as seguintes restrições:

- Tipo de alocação;
- Número de estágios;
- Número, frequência ou seqüência dos padrões;
- Dinâmica de alocação.

O tipo de alocação é uma propriedade fundamental na classificação de um problema de corte e empacotamento. O corte de objetos para produção de itens pode ser visto como uma alocação de itens em objetos, onde duas categorias de alocação se destacam na distinção de um problema:

- i) Típica dos problemas clássicos de corte e empacotamento, a designação de um conjunto de itens a um subconjunto de objetos.
- ii) Observada no carregamento de *pallet*, quando um subconjunto de itens é designado a um conjunto de objetos.

O número de estágios de um problema está relacionado com a quantidade de passos necessários para definição de um padrão. No problema bidimensional, um padrão pode ser obtido por cortes ou empacotamentos, estagiados (com um número pré-determinado de estágios) ou não-estagiados (sem limitação de estágios).

Nos processos produtivos, tanto as conexões estabelecidas entre as etapas, como as tecnologias envolvidas nos processos de alocação podem impor restrições relacionadas à seqüência ou ordem de padrões. Também podem existir limitações quanto ao número de padrões de mesmo tipo ou de tipos distintos.

A alocação de itens em objetos pode ser de natureza dinâmica ou estática. Na alocação estática, se os objetos e itens são previamente conhecidos, o processo é dito *off-line*, caso contrário o processo é *on-line*. Já no processo dinâmico, em função da não disponibilidade de objetos e itens em um mesmo período, as alocações seguem uma regra pré-estabelecida que permite a realocação dos itens em objetos.

2.8. OBJETIVOS

Um objetivo significa usar um critério a ser maximizado ou minimizado para expressar a dimensão da eficácia obtida na solução de um problema. Alguns critérios a serem satisfeitos, em problemas de corte e empacotamento, são listados abaixo:

- i) Minimizar a perda de material nos processos de corte ou empacotamento;
- ii) Minimizar os custos envolvidos no processo produtivo, como por exemplo, despesas com armazenagem;
- iii) Maximizar os lucros com a eficácia e a qualidade dos processos.

2.9. ESTADO DA INFORMAÇÃO E VARIABILIDADE

Determinar se os dados de um problema são determinísticos ou estocásticos, ou ainda, se estes são exatos ou podem ser variáveis, são características relevantes não apenas para os problemas de corte e empacotamento. Por exemplo, uma demanda de determinados itens ordenada em um pedido, geralmente, possui dados determinísticos, podendo ser variável caso certas mudanças sejam admitidas pelos clientes.

2.10. NOMECLATURA DE DYCKHOFF

As características dimensionalidade, sortimento e restrições de alocação constituem uma base para elaboração de uma nomenclatura que associe os problemas de corte e empacotamento afins, além de influenciarem diretamente na escolha e na complexidade do método de solução. A tipologia de um problema é indicada através da quádrupla, dimensionalidade/ tipo de alocação/ sortimento de objetos/ sortimento de itens, e cada característica é subdividida em determinados tipos que são indicados por letras, conforme o quadro 2.1 a seguir:

Quadro 2.1 - Nomenclatura da Tipologia.

Dimensionalidade	Tipo de Alocação
(1) Unidimensional; (2) Bidimensional; (3) Tridimensional; (N) N-dimensional, com $N > 3$.	(B) Todos os objetos e uma seleção de itens; (V) Uma seleção de objetos e todos os itens.
Sortimento de Objetos	Sortimento de Itens
(O) Um objeto; (I) Objetos de formatos idênticos; (D) Objetos de formatos distintos.	(F) Poucos itens de diferentes formatos; (M) Muitos itens de muitos formatos distintos; (R) Muitos itens de formatos distintos em relativa quantidade; (C) Itens de formatos congruentes.

Os agrupamentos formados por todos os tipos destas quatro características indicam 96 tipos distintos de problemas de corte e empacotamento, onde cada agrupamento consta de uma quadrúpla ordenada $\alpha/\beta/\gamma/\delta$, em que α representa o número de dimensões relevantes do problema, β o tipo de alocação considerada com relação a objetos e itens, γ o sortimento dos objetos envolvidos e δ o sortimento dos itens requisitados. A notação $2/V//R$ é utilizada para indicar o problema abordado nesta dissertação, que é do tipo bidimensional (2), com todos os itens sendo designados a uma seleção de objetos (V), objetos estes de formatos idênticos (I) e itens de formatos distintos em relativa quantidade (R). No quadro 2.2 serão listados alguns problemas clássicos de corte e empacotamento encontrados na literatura, com as suas respectivas tipologias, como por exemplo, o Problema de Corte Bidimensional.

Quadro 2.2 - Tipologia de alguns problemas.

Problema	Tipo
Problema da mochila clássico	1/B/O/
Problema da mochila multidimensional	/B/O/
Problema do carregamento de <i>pallet</i>	2/B/O/C
Problema do carregamento de veículos	1/V//F ou 1/V//F
Problema do carregamento de contêiner	3/V// ou 3/B/O/
Problema do <i>bin packing</i> clássico	1/V//M
Problema do <i>bin packing</i> dual	1/B/O/M
Problema do <i>bin packing</i> bidimensional	2/V/D/M
Problema do <i>cutting stock</i> clássico	1/V//R
Problema do <i>cutting stock</i> bidimensional	2/V//R
Problema do <i>cutting stock</i> generalizado	1///, 2/// ou 3///
Problema do balanceamento de uma linha de montagem	1/V//M
Problema de alocação de memória	1/V//M
Problema de alocação de tarefas em multiprocessador	1/V//M
Problema de câmbio monetário	1/B/O/R
Problema de investimento financeiro em multiperíodos	N/B/O/

3. O PROBLEMA DE CORTE BIDIMENSIONAL

Em diversos processos produtivos encontram-se atividades de corte de material, e estes cortes são efetuados em uma peça retangular (objeto) para atender a uma demanda de peças retangulares menores (itens), de maneira que o desperdício de material durante este processo seja minimizado. Este problema caracteriza-se como bidimensional quando duas dimensões são fundamentais na sua resolução, ou seja, a largura e o comprimento das peças envolvidas no processo são essenciais na definição de um padrão de corte.

O problema de corte bidimensional de peças regulares, envolvendo objetos e itens de formato retangular, é freqüentemente abordado na literatura. Porém, em alguns casos, as peças são irregulares, ou seja, o formato destas não é retangular. A figura 3.1 exibirá alguns exemplos de contornos de peças que possuem forma regular e irregular.

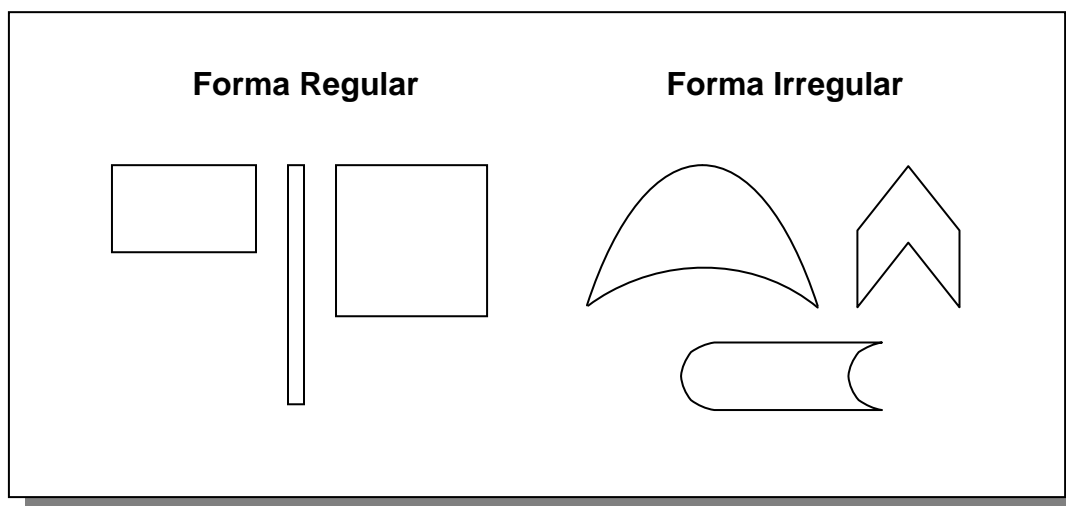


Figura 3.1 - Peças regulares e irregulares.

De acordo com a geometria das peças envolvidas e as características operacionais do problema, se as peças apresentam formato retangular, um padrão de corte pode ser obtido através de cortes guilhotinados ou não-guilhotinados. Estes

padrões guilhotinados são distinguidos em ortogonais e não ortogonais. Se um corte ao ser realizado em um retângulo gerar outros dois retângulos, este é chamado de guilhotinado ortogonal, quando não, é denominado de guilhotinado não ortogonal. A fim de delimitar o problema, os cortes efetivados devem ser paralelos aos lados do retângulo, conforme será apresentado na figura 3.2, e um padrão de corte guilhotinado é aquele definido por série de cortes guilhotinados.

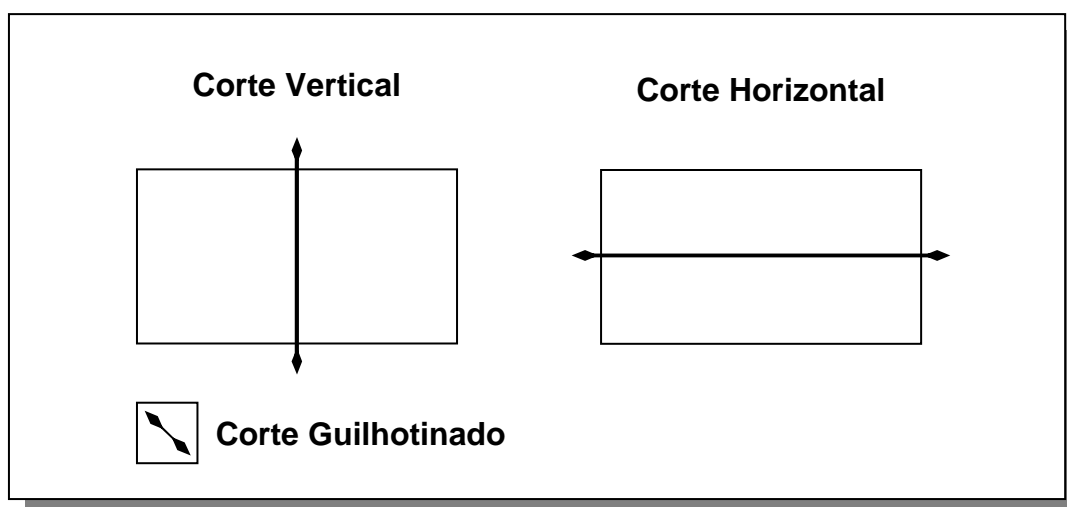


Figura 3.2 - Corte guilhotinado ortogonal.

Na utilização de cortes guilhotinados, estes ainda são classificados em:

- Cortes estagiados;
- Cortes não-estagiados.

Os cortes são estagiados se houver restrições, provenientes das características operacionais do problema, que limitam o número de estágios permitidos para determinação de um padrão de corte. No entanto, se o número de estágios em um padrão de corte é irrelevante na solução de um problema, os cortes são ditos não-estagiados. A figura 3.3 ilustrará um processo de execução de cortes guilhotinados, onde os cortes efetuados em uma única direção definem um estágio e a cada mudança na direção dos cortes, um novo estágio é estabelecido.

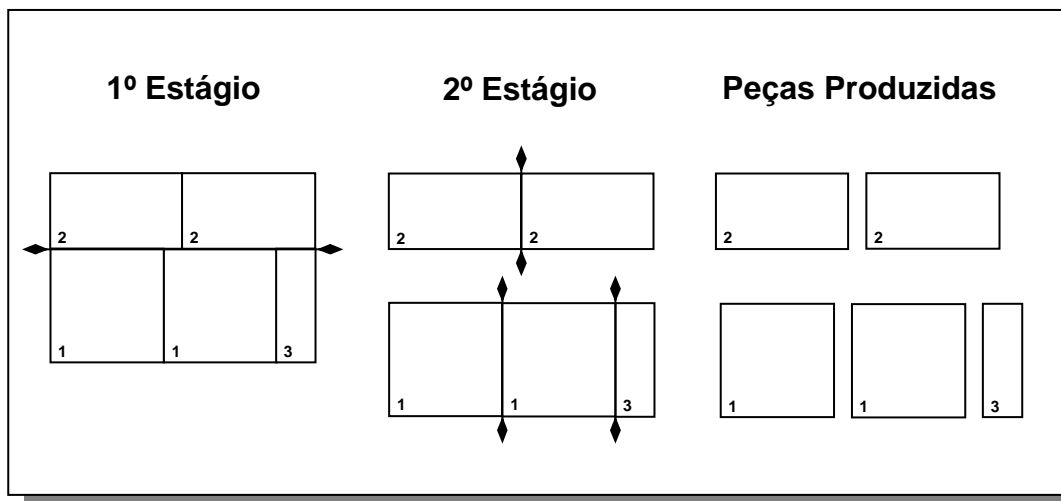


Figura 3.3 - Cortes guilhotinados em dois estágios.

Com relação ao número de peças existentes em um padrão de corte, o problema de corte pode ser restrito ou irrestrito. Se não houver restrições de limitação associadas ao número de peças, o problema é irrestrito. Caso contrário, o problema é dito restrito e a imposição de um limite superior à quantidade de uma determinada peça, em um padrão, acrescenta dificuldades à resolução do problema.

3.1. PROBLEMA DE CORTE BIDIMENSIONAL GUILHOTINADO E RESTRITO

Nessa dissertação é realizado um estudo sobre o problema de corte de peças regulares, com quantidades pré-fixadas, a partir de uma peça maior, também com formato retangular, disponível em estoque. Sendo os cortes efetuados do tipo guilhotinado não-estagiado, este é citado na literatura como o Problema de Corte Bidimensional Guilhotinado e Restrito.

As figuras a seguir ilustram o problema de corte bidimensional restrito, em que os cortes são do tipo guilhotina.

Na figura 3.4, tem-se uma placa retangular com comprimento 200 e largura 100. Supondo a necessidade de realizar cortes em placas retangulares

estocadas, contendo estas dimensões, a fim de produzir peças retangulares menores demandadas.

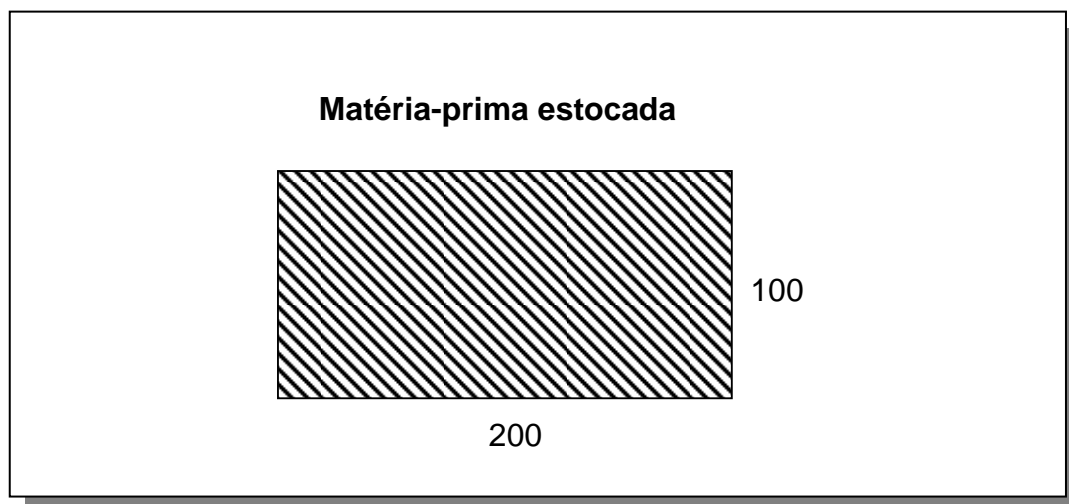


Figura 3.4 - Peças a serem cortadas.

Na figura 3.5, ilustram-se as peças requeridas com suas respectivas dimensões e demandas. As dimensões e demandas são: 50 x 100, 60 x 50, 40 x 40, 40 x 10, e 4, 4, 3, 5, respectivamente.

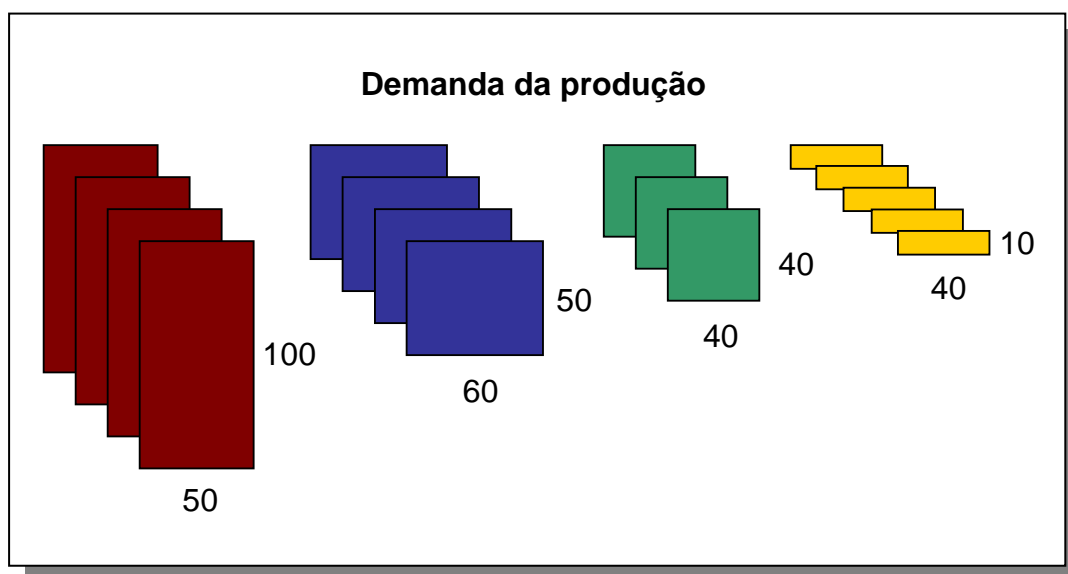


Figura 3.5 - Peças a serem produzidas.

A figura 3.6 apresenta os padrões de corte a serem executados. As peças serão obtidas através de cortes guilhotinados feitos na placa em estoque de forma a minimizar as perdas de matéria-prima.

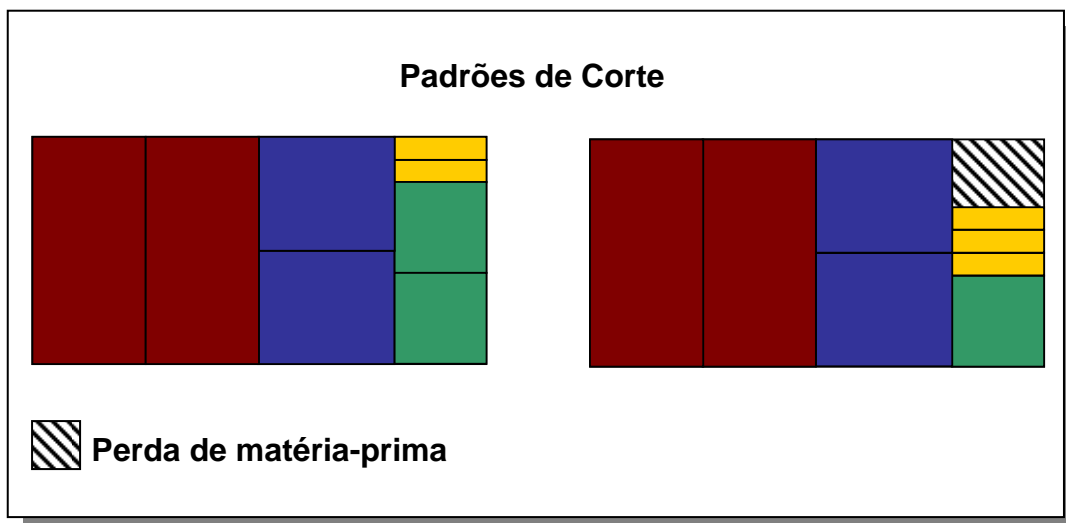


Figura 3.6 - Padrões de corte.

3.2. FORMULAÇÃO DO PROBLEMA DE CORTE BIDIMENSIONAL GUILHOTINADO E RESTRITO

Com os padrões de corte definidos e uma quantidade de placas em estoque suficiente para atender a uma demanda, este problema de corte pode ser formulado como um problema de Programação Linear. Esta formulação consiste em determinar o número de vezes que cada um destes padrões de corte será utilizado, de forma a satisfazer uma demanda de peças pré-estabelecida, consumindo-se o menor número possível de placas estocadas. Sendo assim, a formulação matemática básica do problema de corte restrito apresenta a seguinte estrutura:

$$\begin{aligned}
 \text{Min } Z &= \sum_{j=1}^n x_j \\
 \text{s.a. : } & \sum_{j=1}^n a_{ij} x_j = d_i, \quad i=1, \dots, m \\
 & x_j \geq 0 \text{ e Inteiro, } j=1, \dots, n
 \end{aligned}$$

Onde:

- x_j é a variável que representa o número de vezes que uma peça em estoque é utilizada, segundo o padrão de corte j ;
- a_{ij} é a constante que indica número de itens do tipo i gerados pelo padrão de corte j ;
- d_i é a constante que informa o número de itens do tipo i que deve ser obtido;
- $[a_{1j}, \dots, a_{mj}]^t$ é a coluna j da matriz das restrições que corresponde ao padrão de corte j ;
- m é a constante que indica o número de itens distintos;
- n é a constante que indica o número total de padrões de corte definidos.

A condição que o modelo impõe às variáveis x_j de não assumir valores contínuos torna o problema difícil de ser resolvido, até para problemas relativamente pequenos. Se a quantidade de itens diferentes, encontrados em problemas de médio e grande porte, promove um aumento considerável no número n de padrões viáveis e para encontrar a solução ótima do problema é preciso conhecer todos os padrões de corte, então a utilização do método simplex para resolução deste problema se torna inviável nestas circunstâncias, visto que o número de colunas $[a_{1j}, \dots, a_{mj}]^t$ pode chegar facilmente à classe dos milhões.

Estes obstáculos podem ser contornados, geralmente nos casos em que o número de itens distintos é muito inferior ao encontrado na prática, relaxando as restrições de integralidade, junto a um processo de geração de colunas. Proposto

por Gilmore e Gomory (1961, 1963, 1965), o procedimento inicia-se com o modelo contendo alguns dos muitos padrões de corte viáveis e baseia-se na utilização do método simplex, onde, a cada iteração, é gerado e inserido um novo padrão, desde que melhore o valor da função objetivo. Caso contrário, a solução corrente relaxada é ótima. Sendo o problema de minimização, é conveniente que entre na base a variável x_j que apresenta o custo reduzido $(c_j - z_j)$ mais negativo. Sejam:

$$\begin{cases} c_j = 1 \\ z_j = c_B B^{-1} a_j \Rightarrow z_j = \Pi a_j \Rightarrow z_j = \sum_{i=1}^m \pi_i a_{ij} \end{cases}$$

O objetivo de selecionar a melhor coluna (ou padrão), entre as já existentes e as que vão ser geradas, resulta em:

$$\text{Min} \left(1 - \sum_{i=1}^m \pi_i a_{ij} \right) \Rightarrow \text{Max} \sum_{i=1}^m \pi_i a_{ij}$$

Onde:

- π_i são os multiplicadores para cada restrição i ;
- Π é o vetor dos multiplicadores;
- c_j é o coeficiente da variável j na função objetivo;
- c_B é o vetor das variáveis básicas na função objetivo;
- B^{-1} é a matriz dos coeficientes das variáveis básicas nas restrições do PPL.

Conseqüentemente, a escolha da variável que a cada iteração deve entrar na base decorrerá da resolução do seguinte problema da mochila:

$$\begin{aligned} \text{Max } Z' &= \sum_{i=1}^m \pi_i a_{ij} \\ \text{s.a. : } & \left[a_{1j}, \dots, a_{mj} \right]^t \\ & a_{ij} \geq 0 \text{ e Inteiro} \end{aligned}$$

Onde:

- $[a_{1j}, \dots, a_{mj}]^t$ representa um padrão de corte viável que gera a_{ij} itens do tipo i .

A viabilidade de um padrão de corte depende das restrições geométricas e operacionais envolvidas em cada problema. Em se tratando do problema de corte bidimensional restrito, com cortes guilhotinados e não-estagiados, não se tem a informação de que algum trabalho na literatura tenha apresentado um modelo que gere padrões nestas condições (SILVEIRA, 1999).

A complexidade de se modelar este problema é comprovada em Gramani (*apud* PINTO; ARENALES, 1998), que apresenta um modelo matemático, com restrições não-lineares, para resolução do problema da mochila. Neste trabalho, além da imposição de um limite ao número de peças produzidas (restrito), os padrões são definidos com cortes guilhotinados em dois estágios. A determinação do padrão de corte ótimo para o problema de corte bidimensional restrito em dois estágios é modelado como segue.

Supondo um estoque de placas retangulares (C, L) , com comprimento C e largura L , que dispõe de uma quantidade suficiente de placas para atender a uma demanda d_i de peças retangulares de dimensões (c_i, l_j) , onde $c_i \leq C$, $l_j \leq L$ e $i = 1, \dots, m$. Considere os k padrões unidimensionais viáveis para as faixas (C, l_j) , onde $j = 1, \dots, m$ e $k = 1, \dots, K_j$. Sendo K_j o maior número possível de faixas (C, l_j) produzidas em uma placa (C, L) , isto é,

$$K_j = \left\lfloor \frac{L}{l_j} \right\rfloor$$

Seja o k -ésimo padrão de corte unidimensional para a faixa (C, l_j) designado pelo vetor $(\alpha_{kj}^1, \dots, \alpha_{kj}^m)$, $\alpha_{kj}^i \geq 0$ e inteiro, então:

$$\sum_{i=1}^m \alpha_{kj}^i c_i \leq C$$

No caso de $l_i > l_j$, a componente α_{kj}^i é igual à zero.

Já em relação ao número de faixas (C, l_j) arranjadas em uma placa (C, L), indicado por β_{kj} , deve-se considerar:

$$\sum_{j=1}^m l_j \sum_{k=1}^{K_j} \beta_{kj} \leq L$$

Desta forma, o número de peças produzidas do tipo i é obtido pelo:

$$\sum_{j=1}^m \sum_{k=1}^{K_j} \alpha_{kj}^i \beta_{kj}$$

Como a quantidade de peças produzidas do tipo i não deve superar a demanda, tem-se:

$$\sum_{j=1}^m \sum_{k=1}^{K_j} \alpha_{kj}^i \beta_{kj} \leq d_i$$

Sendo assim, o modelo para determinar o padrão de corte ótimo em um problema de corte bidimensional restrito em dois estágios é apresentado a seguir:

$$\begin{aligned} \text{Max } Z' &= \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^{K_j} \alpha_{kj}^i \beta_{kj} \pi_i \\ \text{s.a.:} \quad &\sum_{i=1}^m \alpha_{kj}^i c_i \leq C, \quad j=1, \dots, m \\ &\sum_{j=1}^m l_j \sum_{k=1}^{K_j} \beta_{kj} \leq L, \quad k_j=1, \dots, K_j \\ &\sum_{j=1}^m \sum_{k=1}^{K_j} \alpha_{kj}^i \beta_{kj} \leq d_i \\ &\alpha_{kj}^i, \beta_{kj} \geq 0 \text{ e Inteiros} \end{aligned}$$

O desejo de produzir o padrão de corte ótimo, para o problema de corte bidimensional guilhotinado e restrito, esbarra na dificuldade de exibir uma formulação matemática que não extrapole as dimensões da placa em estoque e nem apresente superposição das peças (PARADA; PRADENAS; SOLARES, 2000), com a demanda sendo satisfeita por meio de cortes não estagiados.

Desta forma, o problema instituído de determinar um padrão de corte que gere a menor perda de material, depois de obtidas as peças, pode ser descrito por um modelo analítico que apresenta x_i como variável de decisão, onde x_i indica o número de vezes que a peça i está contida no padrão de corte.

Sendo a função objetivo diretamente associada a variável x_i ,

$$\text{Min } Z = CL - \sum_{i=1}^m c_i l_i x_i$$

Esta função objetivo corresponde à área perdida depois da execução dos cortes definidos em um padrão.

Como o problema é restrito, se estabelece um limite superior ao número de peças i no padrão de corte, de modo a gerar um padrão de corte usando não mais do que d_i réplicas da peça i . Assim sendo, a variável de decisão x_i está sujeita a:

$$\sum_{i=1}^n x_i \leq d_i$$

Devido à complexidade de se apresentar uma formulação que trate o tipo de corte executado e garanta a viabilidade do padrão definido, a restrição associada a esta particularidade é expressa desta maneira:

Cortes Guilhotinados Viáveis.

Logo, o modelo analítico que se institui para o problema de se gerar o padrão de corte ótimo, no caso em que os cortes não são estagiados, é exibido a seguir.

$$\text{Min } Z = CL - \sum_{i=1}^m c_i l_i x_i$$

$$\text{s.a.: } \sum_{i=1}^n x_i \leq d_i$$

Cortes Guilhotinados Viáveis

$x_i \geq 0$ e *Inteiro*

Diante da dificuldade de se determinar o padrão de corte ótimo para o problema em questão e da impraticabilidade da técnica de programação linear com geração de coluna para problemas de médio e grande porte, será utilizado um algoritmo aproximado, também denominado heurístico, com o objetivo de viabilizar a obtenção de soluções para o problema corte bidimensional guilhotinado e restrito.

3.3. SOLUÇÕES APROXIMADAS

Os problemas de corte de peças pertencem à classe de problemas NP-*Hard*, onde a utilização de algoritmos exatos para resolução do problema de médio e grande porte não é conveniente. Em geral, os problemas reais apresentam um número elevado de possíveis padrões de cortes a serem gerados e a intenção de enumerá-los, em busca da solução que apresente o melhor aproveitamento, é inviável do ponto de vista prático. Com isso, faz-se necessário o uso de algoritmos aproximados que buscam encontrar boas soluções em tempos de execução razoáveis.

Na prática, o fator tempo é extremamente relevante, assim como uma boa solução. Desta forma, se torna conveniente abrir mão de uma solução ótima, obtida através de um algoritmo exato, devido ao elevado tempo computacional para se alcançar esta solução, por uma solução de boa qualidade, obtida por uma heurística, em um tempo computacional coerente com as decisões a serem tomadas pelos setores produtivos.

A idéia é buscar soluções rápidas para resolver os problemas de corte bidimensional guilhotinado e restrito nas empresas em tempo real. Para isso, utiliza-se uma ferramenta heurística emergente, a GRASP (*Greedy Randomized Adaptive Search Procedure*), para encontrar soluções próximas do ótimo, ou até mesmo o ótimo, de forma rápida.

3.4. REVISÃO BIBLIOGRÁFICA

Diversos pesquisadores têm se dedicado a buscar soluções para os problemas de corte de estoque. Alguns trabalhos, referindo-se ao problema abordado serão citados seguindo uma evolução cronológica.

Em trabalhos pioneiros, Gilmore e Gomory (1961, 1963) propuseram um procedimento de geração de colunas para obtenção da solução ótima de problemas de corte. Empregando técnicas eficientes para resolução do subproblema da mochila, Gilmore e Gomory apresentaram em 1961, um algoritmo de Programação Dinâmica e, em 1963, um algoritmo lexicográfico, ambos para o problema de corte unidimensional.

Em Gilmore e Gomory (1965), o subproblema da mochila foi tratado para o problema de corte bidimensional guilhotinado em dois e três estágios. Neste trabalho, o subproblema é decomposto em duas etapas que são resolvidas pelo algoritmo lexicográfico. Na primeira etapa, são produzidas faixas ótimas, maximizando o valor das peças que são encaixadas em uma faixa com largura l_j ($l_j \leq L$), e na segunda etapa, as faixas são selecionadas e arrumadas em um retângulo (C, L) , com o intuito de gerar um padrão de corte com maior valor possível.

Com respeito à geração de padrões em dois ou três estágios, em dois estágios é imposto que as peças pertencentes às faixas tenham larguras iguais, e em três estágios é necessário apenas que a largura das peças que compõe a faixa (C, l_j) não seja superior a largura l_j da faixa.

Já em Gilmore e Gomory (1967), foi apresentada uma rotina fundamentada em Programação Dinâmica, que gera padrões de corte guilhotinado não estagiado. Os métodos propostos por Gilmore e Gomory são usados nos casos em que o número de vezes que uma peça pode ser produzida em um padrão de corte não está limitado, ou seja, problemas irrestritos.

Herz (1972) propôs uma melhoria na formulação apresentada por Gilmore e Gomory (1966), utilizando algoritmos recursivos para determinar o melhor padrão de corte possível na resolução de problemas bidimensionais irrestritos. Embora alguns pesquisadores tenham feito algumas modificações nos algoritmos propostos por Gilmore e Gomory, ou mesmo desenvolvido algoritmos alternativos (HERZ, 1972) o benefício obtido com as alterações promovidas, segundo Oliveira e Ferreira (1993), não compensa o acréscimo de dificuldade na implementação desses algoritmos, nem suprime o verdadeiro problema do algoritmo de Gilmore e Gomory, ou seja, continuam não sendo praticáveis para problemas de médio ou grande porte.

Utilizando um algoritmo de busca em árvores, que emprega uma rotina baseada no problema de transporte, Christofides e Whitlock (1977) sugeriram um método exato para resolução do problema de corte bidimensional restrito que utiliza programação dinâmica junto a um procedimento para a avaliação de nós. Neste caso, cada nó determinado por um conjunto de itens cortados, a localização do próximo corte a ser efetuado e os ramos da árvore são definidos com as condições impostas para realização de um corte guilhotinado.

Wang (1983) apresenta um algoritmo heurístico que gera um padrão de corte a partir de uma seqüência de construções horizontais e verticais, em estruturas resultantes da composição das peças requeridas. Estas estruturas são produtos do agrupamento das peças na horizontal ou na vertical. Com a finalidade de diminuir o número de subconjuntos de estruturas armazenados, foi determinado um limite para

o percentual de perda encontrada em cada subconjunto. Este artigo é um clássico sobre o problema de corte bidimensional guilhotinado e restrito.

Oliveira e Ferreira (1990) introduzem modificações no algoritmo desenvolvido por Wang, melhorando o desempenho. O Algoritmo de Wang Modificado é o resultado de uma alteração no nível de aspiração, o critério de rejeição de soluções indesejáveis. Desta forma, o algoritmo avalia a perda associada a cada solução parcial construída (padrão embrionário). Caso este valor seja maior que o limite estabelecido, a solução parcial é rejeitada.

Um procedimento de busca em grafos E/OU invertido foi apresentado por Alvarenga e Daza (1992) para o problema de corte bidimensional. A composição entre retângulos, tal como empregado no algoritmo de Wang, foi representada por um grafo E/OU invertido, sendo que cada estrutura equivale a um nó e cada agrupamento entre as estruturas representa um arco. Os nós são agrupados dois a dois para gerar um terceiro, assim um par de nós dá origem a dois novos nós considerando a geração de um agrupamento horizontal e um vertical. Quando comparado ao algoritmo de Wang Modificado, apresentou vantagens, tanto em relação à memória requerida como em número de operações.

Carneiro (1994) propôs resolver o problema de corte bidimensional guilhotinado e restrito através de um Algoritmo Genético. A seleção é feita pelo critério de preenchimento da placa. Em consequência disto, os padrões que apresentarem a menor perda, provavelmente, serão selecionados para constituir a próxima geração. Já o cruzamento se dá em nível dos retângulos menores que formam a placa a ser cortada.

Dois algoritmos heurísticos, Busca Tabu e *Simulated Annealing*, são apresentados por Amaral (1994) para tratar o problema de corte bidimensional restrito, com cortes guilhotinados ortogonais. Com o objetivo de produzir padrões, de uma forma ótima, esta abordagem permite aceitar padrões de corte inviáveis como soluções na busca do melhor padrão. Sendo assim, uma penalidade é inserida na função objetivo, de forma a conduzir o procedimento para uma solução viável. Para ilustrar a efetividade dos algoritmos propostos, foram apresentados resultados

computacionais, descrevendo seus respectivos desempenhos para alguns exemplos gerados aleatoriamente e outros extraídos de trabalhos considerados na literatura. As instâncias, assim como os resultados apresentados pelos algoritmos Busca Tabu e *Simulated Annealing*, são utilizadas nessa dissertação para validar o algoritmo proposto.

Christofides e Hadjiconstantinou (1995) apresentam algoritmo de busca em árvores para resolução do problema de corte bidimensional guilhotinado e restrito, que é um melhoramento do algoritmo proposto por Christofides e Whitlock (1977). Este algoritmo exato diminui o espaço de soluções a ser examinado, utilizando um limite superior proveniente de uma Relaxação do Espaço de Estados de uma formulação de programação dinâmica para o problema. Um método de modificação do espaço de estados, denominado Espaço de Estados Ascendentes, associado com a recursão relaxada, é utilizado na otimização dos limites superiores. Os resultados obtidos indicam que este procedimento executa razoavelmente bem problemas de porte médio e apresenta um desempenho computacional bem superior à proposta de Christofides e Whitlock.

Em Daza, Alvarenga e Diego (1995) foi proposto um algoritmo exato para tratar o problema de corte bidimensional guilhotinado e restrito. Uma generalização dos algoritmos apresentados por Wang (1983) e Oliveira e Ferreira (1990) foi usada para definir o algoritmo AAO^{*}, que é um método de busca sobre grafos E/OU aditivo. Estes grafos são usados para representar uma combinação entre as estruturas, garantindo a viabilidade do padrão a ser gerado. O algoritmo AAO^{*} usa informação futura para orientar a busca no espaço de soluções. Para isto, uma função é definida como a medida aproximada da perda futura associada a uma solução parcial. Fazendo uso desta função, denominada função heurística, é possível analisar várias alternativas percorrendo o domínio do problema parcialmente. Este método exato é um resultado importante deste estudo porque, por meio de uma representação adequada, pode ser usado não só para resolver problemas de corte, mas também resolver problemas tão difíceis quanto este.

Uma abordagem em grafo E/OU para o problema de corte bidimensional restrito estagiado foi proposta por Morabito e Arenales (1996). Esta

proposta consiste em representar padrões de corte como caminhos completos no grafo E/OU, onde os nós representam os retângulos e os arcos representam os cortes guilhotinados, respeitando o número máximo de estágios. Além do fato de que a abordagem em grafo E/OU pode controlar restrições importantes, este procedimento pode ser facilmente estendido para resolver problemas de corte e empacotamento com dimensões múltiplas.

Em um trabalho pioneiro, Vieira Neto (1999) utilizou a técnica GRASP na resolução de problemas de corte unidimensional. Como base para o desenvolvimento do algoritmo GRASP, foi utilizada a técnica heurística FFD (*First Fit Decreasing*). Na implementação da fase de construção do algoritmo proposto por Vieira Neto, foi criada a estratégia de compor a Lista Restrita de Candidatos aplicando o parâmetro de aleatoriedade α ao valor β , que é dado por uma função gulosa que retorna o comprimento do maior item com demanda não atendida. Utilizando dados de problemas reais e de problemas encontrados na literatura, foram executados testes computacionais que permitiram a comparação entre a FFD e a GRASP, mostrando a eficiência do algoritmo GRASP para otimização de cortes unidimensionais.

Cung, Hifi e Le Cun (2000) desenvolveram um algoritmo *Branch-and-Bound*, que é uma nova versão do algoritmo proposto por Hifi (1997) para problemas de corte bidimensionais restritos. Para melhorar o desempenho do algoritmo, foi aumentado o limite inferior inicial, diminuindo inicialmente o espaço de busca. Também se tentou aperfeiçoar o limite superior efetuado a cada nó desenvolvido na árvore, aplicando algumas combinações simples e eficientes. Além disso, foram introduzidas novas estratégias simétricas usadas para negligenciar alguns padrões de corte duplicados.

Inspirados na busca de um melhor desempenho computacional dos métodos de busca orientada, Parada, Pradenas e Solar (2000) sugeriram um método híbrido para resolução dos problemas de corte bidimensional restrito guilhotinado e não-guilhotinado. O algoritmo reuniu elementos construtivos, dos métodos de busca informada, e elementos evolutivos, dos algoritmos genéticos. A proposta consiste em utilizar ramificações reguladas na geração dos nós

intermediários, armazenando a populações de nós que evoluíram, segundo os princípios envolvidos nos algoritmos genéticos.

Em Valdés, Parajón e Tamarit (2002) foi desenvolvido um sofisticado algoritmo Busca Tabu, para problemas de grande porte, que obtém resultados de alta qualidade em tempos computacionais moderados. Também foi desenvolvido um procedimento GRASP que é muito rápido e retorna bons resultados, tanto para problemas restritos, como irrestritos. Para auxiliar na construção dos padrões de corte, foram propostos dois algoritmos heurísticos, baseados em limites superiores simples (BK_1 e BK_2) que são obtidos resolvendo um problema de mochila. A fase de melhoria está focalizada na fusão de cada retângulo desperdiçado, isto se possível, com uma peça adjacente, para criação de um novo retângulo que poderia ser cortado com maior valor. Neste processo, dois retângulos são considerados adjacentes se eles tiverem um lado comum. O critério de parada utilizado pelo algoritmo GRASP é um determinado número de iterações sem aperfeiçoar a melhor solução conhecida.

Vieira Neto (2004) apresentou três algoritmos baseados na técnica GRASP, sendo que dois foram desenvolvidos com a metodologia GRASP com filtro, para o problema de corte unidimensional. Este trabalho teve como escopo avaliar a independência entre as soluções finais e as soluções iniciais construídas, assim como a influência do parâmetro α de aleatoriedade nas soluções finais. Os algoritmos foram testados computacionalmente, utilizando instâncias geradas aleatoriamente e instâncias práticas retiradas da indústria, e apresentaram resultados superiores aos obtidos pela heurística FFD.

4. MÉTODOS HEURÍSTICOS

Otimizar o uso do material a ser cortado, minimizando as perdas do material, pode ser encarado como um problema de otimização combinatória e solucionado a partir de técnicas de Programação Matemática. Os métodos utilizados para resolver problemas de otimização combinatória, dependendo da qualidade da solução que apresentam, podem ser exatos ou heurísticos. Algoritmos exatos são algoritmos enumerativos que examinam todas as soluções (enumeração explícita) ou uma parte destas (enumeração implícita) em busca da solução ótima do problema. Já nos algoritmos heurísticos, não há certeza de se encontrar a solução ótima, entretanto, são executados em tempo polinomial.

O problema de corte bidimensional pertence à classe de problemas denominada *NP-Hard*, onde não se conhece a existência de algoritmos exatos que resolvam este problema em tempo polinomial, ou seja, com baixo esforço computacional, tornando a utilização de métodos enumerativos não conveniente. Isto se deve, ao grande número de arranjos viáveis, encontrados na maioria dos problemas reais, que devem ser comparados para se determinar um arranjo ótimo, acarretando em um elevado tempo computacional na determinação desta solução ótima. Sendo assim, para problemas de médio e grande porte, faz-se aceitável a utilização de algoritmos heurísticos (CAMPELLO; MACULAN, 1989), abrindo mão da solução ótima, quando este retorna uma solução viável de boa qualidade obtida em tempo polinomial, com um valor suficientemente próximo do exato.

Entre as técnicas encontradas na literatura para solução de problemas de otimização combinatória, destacamos neste trabalho as heurísticas.

Uma heurística é uma técnica que busca alcançar uma boa solução utilizando um esforço computacional considerado razoável, sendo capaz de garantir a viabilidade ou a otimalidade da solução encontrada ou, ainda, em muitos casos, ambas, especialmente nas ocasiões em que essa busca partir de uma solução viável próximo ao ótimo (GOLDBARG e LUNA, 2000, p.244).

As heurísticas diferem segundo a estratégia que usam para buscar e construir suas soluções. As heurísticas de construção, como o próprio nome sugere, constroem uma solução adicionando componentes individuais, gradativamente, até encontrar uma solução viável. Destacamos os algoritmos gulosos ou míopes nas heurísticas construtivas, pois constroem uma solução considerando a cada iteração a melhor decisão um passo a diante (CAMPELLO; MACULAN, 1994). As heurísticas construtivas, em muitos casos, são aplicadas para se obter uma solução inicial que poderá ser melhorada por outras heurísticas. Quase sempre esta solução inicial retornada pode ser melhorada e, para isto, emprega-se um artifício de melhoria sobre ela a fim de encontrar um resultado mais interessante. As heurísticas de melhoria ou Busca Local são técnicas baseadas na noção de vizinhança. A Busca Local começa a partir de uma solução inicial viável, que pode ser obtida por uma heurística construtiva ou produzida aleatoriamente, da qual se gera uma vizinhança de soluções, ou seja, conjunto de soluções obtidas a partir de modificações feitas na solução inicial, visando escolher o melhor vizinho para ser a nova solução corrente, seguindo um critério de escolha. Nas heurísticas construtivas o processo é interrompido quando encontra uma solução, já na Busca Local, o processo é repetido até que o critério de parada seja atendido. Porém, estas heurísticas se limitam a problemas específicos e retornam sempre à mesma solução quando partidas de um mesmo ponto (NORONHA, 2001). Métodos mais flexíveis e de caráter geral, apresentando condições de escapar de ótimos locais, são conhecidos na literatura como Metaheurísticas.

As metaheurísticas são métodos mais eficazes de busca local que possibilitam encontrar soluções melhores. Executam procedimentos de busca em vizinhanças, que evitam a parada prematura em ótimos locais durante a procura de soluções de melhor qualidade, aumentando as chances de se chegar ao ótimo global, podendo até mesmo usar uma estratégia de piorar as soluções, a fim de escapar do ótimo local. Este processo respeita uma seqüência de passos bem definidos, até atingir um critério de parada.

As propostas de metaheurísticas encontradas na literatura apresentam diferentes características nas suas estruturas, que distinguem uma das outras (SOUZA, 2002). Essencialmente, estas diferenças se devem ao modo pelo qual é:

- feita a escolha de uma solução inicial;
- determinada a vizinhança de uma solução;
- selecionada uma solução vizinha;
- definida sua parada.

As metaheurísticas envolvidas nesta dissertação de mestrado são *Simulated Annealing*, Busca Tabu e GRASP.

4.1. SIMULATED ANNEALING

O *Simulated Annealing* é uma técnica de busca local probabilística, inicialmente apresentado por Kirkpatrick em 1983, que insere conceitos de mecânica estatística dentro do mundo da otimização combinatória. Fundamentado em uma relação de semelhança com a termodinâmica, quando se simula o resfriamento, gradual e vagaroso, de um conjunto de átomos, previamente perturbados com aquecimento de um sólido. Se o resfriamento for muito rápido, seus átomos formarão uma estrutura irregular e enfraquecida. Este processo físico é conhecido como recozimento (KIRKPATRICK; GELATT; VECCHI, 1983).

A busca local começa a partir de uma solução inicial qualquer, que a cada iteração determina, aleatoriamente, um único vizinho da solução atual. Se o valor obtido com este vizinho, a partir de uma função objetivo, for melhor que o valor da solução atual, o *Simulated Annealing* aceita a solução vizinha e a torna solução atual. Caso contrário, esta solução poderá ser aceita com certa probabilidade, controlada por um parâmetro T (Temperatura) que estabelece a probabilidade de aceitar estas soluções piores. Este procedimento de aceitar soluções piores diminui gradativamente, conforme a Temperatura se aproxima de zero, até o ponto em que nenhuma solução pior é aceita encontrando-se um ótimo local.

A capacidade de o *Simulated Annealing* aceitar movimentos indesejáveis é que o torna capaz de convergir para soluções bem próximas da ótima (DIAZ et al, 1996; NORONHA, 2001; SOUZA, 2002).

4.2. BUSCA TABU

O Busca Tabu é um tipo de busca local com procedimentos determinísticos, proposto inicialmente por Glover (1986), que procura registrar e usar de forma inteligente as informações sobre a história da busca, a fim de realizar decisões mais eficientes no percurso em direção ao ótimo.

O método começa a partir de uma solução inicial viável e para cada solução corrente do problema gera-se uma vizinhança. Uma pesquisa é feita para determinar o melhor vizinho, ou melhor, uma função objetivo avalia as soluções vizinhas e um movimento para melhor solução vizinha é feito. Para evitar ciclos e orientar o algoritmo na fuga de ótimos locais é definida uma memória de curta duração contendo movimentos proibidos, chamada de Lista Tabu. A Lista Tabu contém um número X pré-definido de soluções vizinhas obtidas nas X últimas iterações, que não podem ser visitadas momentaneamente. As soluções proibidas são denominadas Soluções Tabu e os movimentos que levam a estas soluções são denominados Movimentos Tabu. É consentido retirar o Status Tabu de um movimento, se a solução vinculada a este movimento apresenta um valor na função objetivo melhor que o valor da solução atual, e isto é feito através de um critério denominado Função de Aspiração (GLOVER, 1997).

O processo pode ser interrompido, definindo-se um número de iterações a partir do momento em que não se obtém modificações no valor da melhor solução.

A imposição de uma Lista Tabu e uma Função de Aspiração junto com o fato de aceitar a melhor solução vizinha, mesmo que esta solução piore o valor da função objetivo, permite a esta metaheurística fugir de ótimos locais em direção ao

ótimo global, encontrando resultados eficazes (DIAZ et al, 1996; NORONHA, 2001; SOUZA, 2002).

4.3. GRASP

A metodologia GRASP (*Greedy Randomized Adaptive Search Procedure*), surgida na década de 80 e desenvolvida originalmente por Feo e Resende (1989), é essencialmente a combinação de uma heurística construtiva com uma de Busca Local. O método GRASP é um procedimento iterativo probabilístico que atua por amostragem em um espaço de soluções subtraído. A cada iteração é gerada uma solução de forma míope, aleatória e adaptativa que será sujeitada à Busca Local, sendo a melhor solução global mantida como resultado final. Um pseudocódigo genérico GRASP é apresentado a seguir.

Procedimento GRASP ()

- 1 Entrada de Dados ();
 - 2 Para (*critério de parada GRASP não satisfeito*) faça
 - 3 ConstruirSoluçãoGulosaAleatória (*solução*);
 - 4 BuscaLocal (*solução, Viz(solução)*);
 - 5 AtualizarSolução (*solução, melhor solução encontrada*);
 - 6 Fim-para;
 - 7 Retorna (*melhor solução encontrada*);
- Fim GRASP**

A linha 1 do pseudocódigo corresponde à entrada de dados do problema. Entre as linha 2 e 6 temos um conjunto de instruções que são executadas repetidamente até que uma condição específica seja satisfeita. A fase de construção da GRASP é indicada na linha 3, enquanto a linha 4 é a fase de Busca Local. Caso a Busca Local tenha obtido alguma melhoria na solução corrente, a solução é atualizada na linha 5 (FEO; RESENDE, 1995). Todo este processo pode ser interrompido definindo-se um número máximo de iterações, determinando um tempo

máximo de execução ou quando a solução procurada for obtida. Nesta dissertação, assim como na maioria dos trabalhos encontrados na literatura, o critério de parada é estabelecido por um número máximo de iterações.

Como o algoritmo GRASP constitui-se a cada iteração de duas fases, a primeira de construção e a segunda de melhoria, estas são descritas separadamente como segue.

4.3.1. FASE DE CONSTRUÇÃO DA GRASP

Basicamente, na primeira fase, uma solução viável é construída iterativamente apreciando um elemento de cada vez. Este elemento é retirado aleatoriamente de um conjunto restrito denominado de lista restrita de candidatos (LRC). O conjunto LRC é composto pelos elementos mais interessantes de uma lista constituída de candidatos a serem incluídos em uma solução, seguindo um critério guloso. Esta técnica de escolha permite que soluções distintas sejam produzidas em cada iteração GRASP. A cada elemento incluído na solução, adaptações relacionadas às informações do problema são feitas sob a orientação de uma função gulosa. Esta função gulosa mede o benefício, associado à admissão de cada elemento a uma determinada solução. O pseudocódigo descrito abaixo apresenta a fase de construção da GRASP.

Procedimento ConstruirSoluçãoGulosaAletória (*solução*);

```

1  solução = { };
2  Para (solução construída incompleta) faça
3      CriarLRC (LRC);
4      s = ElementoSelecioneadoAleatoriamente (LRC);
5      solução = solução  $\cup$  {s};
6      FunçãoAdaptaçãoGulosa (s);
7  Fim-para;
Fim ConstruirSoluçãoGulosaAletória

```

A construção iterativa de uma solução é iniciada na linha 1 do pseudocódigo. Os comandos entre as linhas 2 e 7 são repetidos até que a solução seja construída. Na linha 3, a lista restrita de candidatos é construída. Um candidato de LRC é selecionado, ao acaso, na linha 4 e acrescentado a solução na linha 5. Na linha 6, a função gulosa atualiza as informações de acordo com o elemento incluído.

Um parâmetro α determina a quantidade de elementos incluídos no LRC, assumindo valores no intervalo $[0,1]$. Seja P um conjunto finito formado pelos próximos elementos p_i a serem incluídos na solução, $f: P \rightarrow \mathbb{R}$ uma função gulosa e $\beta = \max \{ f(p_i); p_i \in P \}$ o melhor valor retornado pela função gulosa. O subconjunto LRC, formado pelos melhores elementos de P ($P \supseteq \text{LRC}$), é determinado aplicando o parâmetro α ao valor β na definição de um intervalo de valores, isto é,

$$\text{LRC} = \{ p_i \in P \mid \alpha \cdot \beta \leq f(p_i) \leq \beta \}$$

Para $\alpha=1$ se gera soluções totalmente gulosas e para $\alpha=0$ se produz soluções totalmente aleatórias. Assim, o parâmetro α regula o grau de miopia e aleatoriedade da fase de construção. Este é o principal parâmetro a ser ajustado no algoritmo GRASP, pois se a cardinalidade de LRC for pequena, menor será o espaço de soluções examinado e, conseqüentemente, a probabilidade de escapar de um ótimo local diminuirá. Já um LRC que apresenta uma cardinalidade grande, produz muitas soluções diferentes, aumentando a perspectiva de escapar de um ótimo local indesejado. Isto implicaria um acréscimo considerável no número de iterações com soluções ruins, assim como implicaria em maiores vizinhanças exploradas. Sendo assim, é imprescindível a análise sobre o valor assumido pelo parâmetro α com respeito à qualidade das soluções encontradas, número de iterações utilizadas e vizinhança explorada (VIEIRA NETO, 2004).

Esta fase termina quando se obtém uma solução e, dependendo da qualidade desta solução, ou seja, quanto mais próxima estiver de um ótimo local aceitável ou do ótimo global, aumenta-se a garantia de sucesso na segunda fase.

4.3.2. FASE DE MELHORIA DA GRASP

A segunda fase é tipicamente um procedimento de Busca Local. As soluções construídas pela primeira fase não, necessariamente, são desejáveis e quase sempre a busca local pode melhorar esta solução. Determinada uma estrutura de vizinhança V para o problema X , que gere um subconjunto de soluções $V(x_i)$ a partir de uma solução x_i do problema. A condição para que uma solução x_i seja declarada como um ótimo local é não haver solução melhor em $V(x_i)$. Na tentativa de melhorar as soluções originadas na fase de construção, emprega-se o procedimento de Busca Local, que examina o espaço de soluções mudando de uma solução para outra vizinha. A seguir é apresentado o pseudocódigo de um procedimento básico de Busca Local para esta fase da GRASP.

```

Procedimento BuscaLocal ( $X, V(X), solução$ );
1   Para (solução não é localmente ótima) faça
2       Encontrar uma melhor solução  $x \in V(solução)$ ;
3        $solução = x$ ;
4   Fim-para;
5   Retorna (solução como ótimo local para  $X$ );
Fim BuscaLocal
  
```

O laço iniciado na linha 1 é executado até encontrar um ótimo local para uma vizinhança. Na linha 2 são efetuados movimentos, isto é, modificações que transformam uma solução em outra pertencente em sua vizinhança, com o intuito de obter um ótimo local. Caso uma solução encontrada na vizinhança seja melhor que a solução tratada, este vizinho passa a ser a solução corrente na linha 3. Finalmente na linha 5, este procedimento finaliza retornando um ótimo local, ou melhor vizinho, para o problema. A condição de parada deste procedimento é não haver solução melhor que a solução atual, em sua vizinhança.

A rapidez com que o processo de busca local encontra uma solução melhor é diretamente proporcional à qualidade da solução gerada na fase de construção. Baseando-se na construção eficiente de uma solução inicial, pode-se conseguir uma redução no número de passos necessários para se alcançar uma solução desejável na fase de melhoria e, conseqüentemente, a diminuição do tempo computacional consumido no processamento desta solução (DIAZ et al, 1996; NORONHA, 2001; SOUZA, 2002).

Uma particularidade que diferencia a metodologia GRASP das metaheurísticas Busca Tabu e *Simulated Annealing*, onde a cada iteração tira-se proveito das informações obtidas nos passos anteriores, é o fato de que as iterações, assim como as soluções geradas por estas, não apresentam vínculos, ou seja, as soluções encontradas no seu processo de busca são completamente independentes.

5. IMPLEMENTAÇÃO DO ALGORITMO GRASP

Considere uma placa retangular estocada $R = (C, L)$ de comprimento C e largura L , em quantidade suficiente, e um conjunto $P = \{(c_1, l_1), (c_2, l_2), \dots, (c_m, l_m)\}$ de peças retangulares menores de comprimento c_i e largura l_i , respectivamente, com demandas d_i a serem atendidas, $i = 1, \dots, m$. Estes elementos podem ser associados a pontos no sistema cartesiano já que a placa e as peças são representadas por pares ordenados. Vê-se assim que nas coordenadas cartesianas, a execução de uma série de cortes guilhotinados paralelos aos eixos coordenados, isto é, aos lados da placa R , para produção de peças i , definiria um padrão para o problema de corte bidimensional guilhotinado e restrito.

Uma peça (c_i, l_i) pertencente a P , alocada no canto esquerdo inferior de $R = (C, L)$, determina dois tipos de corte guilhotinado a serem executados sobre uma placa R , corte horizontal e corte vertical, conforme ilustrado na figura 5.1.

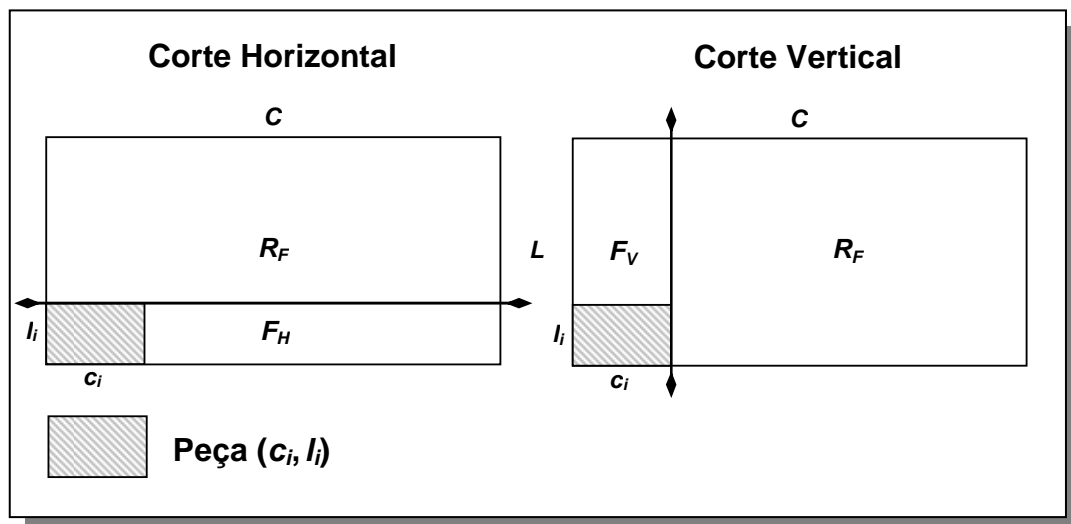


Figura 5.1 - Cortes guilhotinados e faixas guilhotina.

Observa-se na figura 5.1 que tanto no corte horizontal, quanto no corte vertical, são gerados dois novos retângulos. Com a execução de um corte horizontal,

a peça (c_i, l_i) deu origem ao retângulo $F_H = (C, l_i)$ que é denominado de faixa guilhotina horizontal e um sub-retângulo $R_F = (C, L - l_i)$ a ser utilizado, possivelmente, na produção das demais faixas que irão compor o padrão de corte. Na ilustração analisada, um corte efetuado na vertical produz o retângulo $F_V = (c_i, L)$, denominado de faixa guilhotina vertical e outro sub-retângulo $R_F = (C - c_i, L)$ a ser explorado na definição de outras faixas.

Diante disso, um padrão é distinguido pelo conjunto de faixas guilhotinas que o configura e pela maneira como estas faixas são arranjadas. Conforme observado, uma faixa guilhotina é constituída pelo agrupamento de peças na horizontal ou na vertical. As peças p_i incluídas em uma faixa podem apresentar dimensões (c_i, l_i) em relação a um eixo de coordenadas ortogonais e dimensões (l_i, c_i) quando ocorrido uma rotação de 90 graus em relação a este eixo de coordenadas.

Outra questão importante a ser considerada é que, ao executar uma faixa guilhotina horizontal simbolizada por H e uma faixa guilhotina vertical por V , uma peça (c_i, l_i) simbolizada por P e uma peça (l_i, c_i) por R , pode-se descrever uma faixa guilhotina, com as suas respectivas peças, através de uma expressão definida a partir destas letras. Por exemplo, na figura 5.2 a expressão $H3P1$ representa uma faixa guilhotina horizontal contendo três peças p_1 . Ainda na mesma figura, a expressão $H3R1$ indica que a faixa guilhotina horizontal possui três peças p_1 rotacionadas. Da mesma forma, uma faixa guilhotina vertical composta por quatro peças p_2 é indicada pela expressão $V4P2$. Logo, com quatro peças p_3 rotacionadas sua expressão é $V4R2$. Observe que estas expressões dão origem a uma seqüência de caracteres alfanuméricos. Tais caracteres começam sempre por H ou V e se apresentam seguidos de: um dígito representando a quantidade de peças incluídas, um caractere P ou R indicando a orientação destas peças e outro dígito, informando o índice desta peça inserida.

Em seguida, caso uma faixa apresente outras peças, além da peça comprometida com sua geração, novas *strings* de caracteres, agora iniciadas por M , são concatenadas a estas, iniciadas por H ou V . A letra M caracteriza uma melhoria

efetuada em uma faixa guilhotina. Esta situação é apresentada na figura 5.2 pela faixa guilhotina horizontal $H3P1M3P2$.

Logo, um padrão de corte fica caracterizado pelas diferentes expressões apresentadas por cada faixa guilhotina, considerando a ordem de registro destas expressões. O padrão de corte exemplificado na figura 5.3 pode ser descrito pela *string* de caracteres $H2P3M2R5M1P4M1P6M1R7 V2R2 H1P1$.

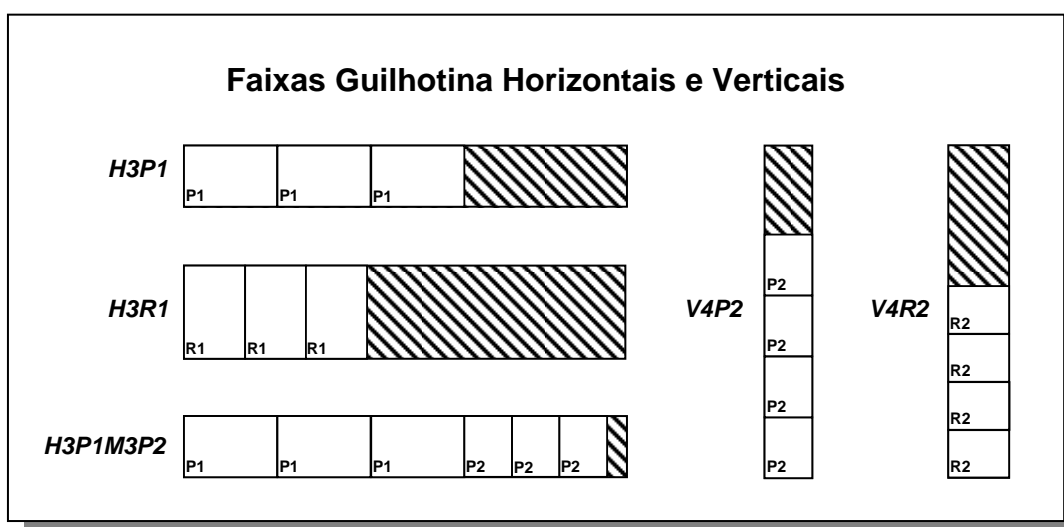


Figura 5.2 - Faixas guilhotina horizontais e verticais.

O presente estudo esclarece que os padrões de corte gerados pelo algoritmo GRASP, segundo a definição apresentada por Wang (1983), possuem uma forma normalizada equivalente. Toda faixa contida no padrão, assim como as peças que a compõe, devem ser posicionadas no lado esquerdo, o mais baixo possível.

É importante ressaltar também que a perda produzida por um padrão de corte é distinguida neste trabalho da seguinte forma: se os sub-retângulos $R_F = (C_F, L_F)$, $C_F \leq C$ e $L_F \leq L$, produzidos por ambos os cortes, vertical e horizontal, não são aproveitados na geração de novas faixas guilhotina, estes passam a configurar como uma perda externa em um padrão de corte. Da mesma forma, uma faixa guilhotina também pode gerar pedaços improdutivos, chamados de perda

interna. Para exemplificar, tem-se a figura 5.3 destacando os dois tipos de perda em um padrão de corte.

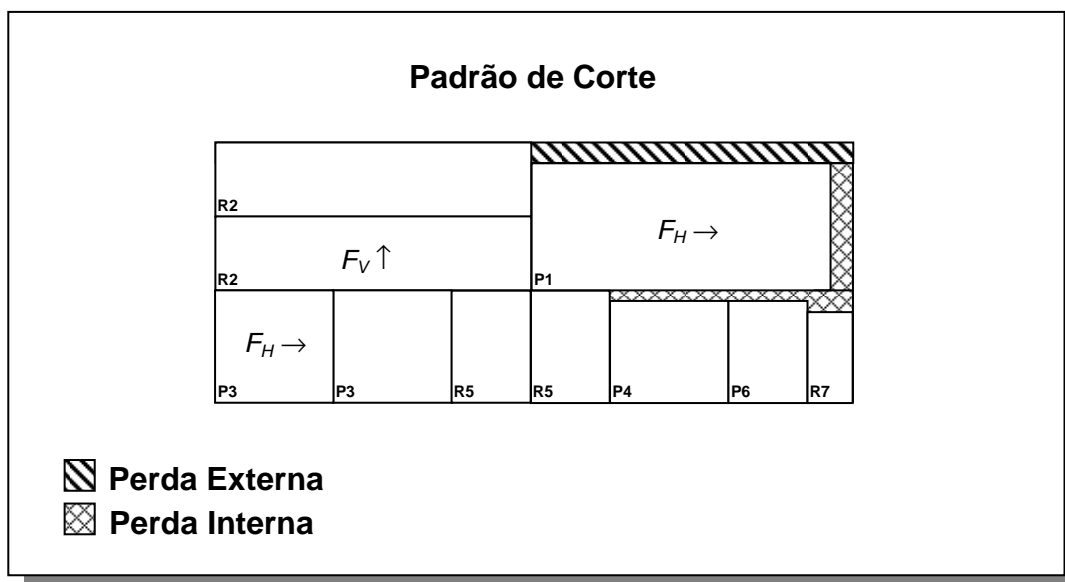


Figura 5.3 - Perda Externa e Perda Interna.

Sendo assim, o valor da perda interna e externa, em um padrão de corte, é definido como a medida da superfície da placa R não aproveitada dentro de e fora de uma faixa, respectivamente.

Em síntese, uma solução para o problema de corte bidimensional guilhotinado e restrito é definida por um conjunto de padrões de corte, que são representados, respectivamente, por *strings* de caracteres. A perda total encontrada na solução equivale à soma das perdas externas e internas produzidas nos padrões de corte que a compõe. Considera-se a melhor solução aquela que apresenta a menor perda, ou melhor, o maior percentual de aproveitamento em relação ao somatório de placas utilizadas. No algoritmo descrito a seguir, o valor da função objetivo para uma solução é igual ao valor encontrado no percentual de aproveitamento da mesma.

5.1 APLICAÇÃO DA GRASP AO PROBLEMA DE CORTE BIDIMENSIONAL

Nem sempre é possível achar uma solução ótima para um problema pertencente à classe *NP-Hard*. Caso seja determinada a solução ótima de um problema de corte bidimensional guilhotinado e restrito, certamente, esta apresentaria faixas otimizadas em padrões de corte bem arranjados. Em decorrência desta dedução, é proposto um algoritmo, baseado na técnica GRASP, para solução do problema de corte bidimensional guilhotinado e restrito. O procedimento GRASP-2D, implementado neste trabalho, atua diretamente no processo de formação e organização de faixas guilhotina em um padrão de corte, decompondo a resolução do problema em vários problemas unidimensionais.

Nos itens 5.1.1 e 5.1.2, têm-se o intuito de apresentar as duas fases características da metodologia GRASP aplicadas ao problema de corte bidimensional guilhotinado e restrito.

5.1.1 FASE DE CONSTRUÇÃO DO ALGORITMO GRASP-2D

É importante esclarecer que na fase de construção do GRASP-2D, faixas guilhotinas viáveis F_H e F_V são construídas, iterativamente, utilizando uma peça p_i de cada vez. Sendo assim, tem-se nesse processo três elementos fundamentais: uma lista C' construída pelas próximas peças candidatas a determinar uma faixa, uma função gulosa $v: P \rightarrow \mathbb{R}$, que para cada peça $p_i \in P$ associa-se um único valor igual à medida de sua área ($v_i = c_i \cdot l_i$) e um critério guloso $\beta = \max \{v(p_i); p_i \in C'\}$ que utiliza, inicialmente, as maiores peças na definição das faixas. Seja $\alpha \in [0,1]$ o parâmetro que determina a cardinalidade de LRC , uma peça é escolhida de forma aleatória em um conjunto

$$LRC = \{p_i \in C' \mid \alpha \cdot \beta \leq v(p_i) \leq \beta\},$$

constituído das melhores peças encontradas em C' .

A peça resultante da escolha aleatória em *LRC*, chamada de p_k , vai construir duas faixas guilhotinas viáveis a serem geradas por um corte guilhotinado horizontal e outro vertical. Nas duas faixas construídas, F_H e F_V , são calculados a quantidade de peças p_k incluídas e o valor da perda interna produzida. É também determinado, ainda, para cada faixa construída, um sub-retângulo referente à sua respectiva perda interna.

Vale lembrar que, se a inclusão da peça selecionada em *LRC* não for viável na geração do padrão de corte, esta será excluída de C' e as informações relacionadas ao problema serão atualizadas para a escolha de um próximo elemento de *LRC*. Caso a peça escolhida p_k possa ser incluída, tanto com uma rotação de 90 graus (r_k) quanto sem a rotação permitida (p_k), uma escolha aleatória vai definir a orientação assumida pela peça p_k .

5.1.2 FASE DE MELHORIA DO ALGORITMO GRASP-2D

As duas faixas, F_H e F_V , produzidas pela fase de construção do GRASP-2D, não necessariamente, foram bem aproveitadas. A fase de melhoria tem como objetivo aprimorar a utilidade das faixas guilhotinas construídas, minimizando o valor percentual da perda gerada em cada faixa. Para intensificar este processo, dois conjuntos B_H e B_V são criados com a finalidade de excluir movimentos desnecessários que acarretariam em faixas guilhotinas ineficazes, ou fariam descaracterizar o corte do tipo guilhotinado. As peças $p_i = (c_i, l_i)$ pertencentes à B_H e B_V , peças candidatas a efetuar uma melhoria nas respectivas faixas F_H e F_V , não devem possuir medidas c_i e l_i , que extrapolem as dimensões da faixa, ou até da placa, quando incluídas. Assim, com a criação destes conjuntos é possível acelerar a fase de melhoria deste procedimento.

A princípio, o conjunto B_H é constituído das peças de C' que possam ser incluídas no sub-retângulo associado à perda interna inicial da faixa F_H . O item mais interessante deste conjunto é selecionado de acordo com a medida apresentada pela sua altura, ou seja, é escolhida a peça p_i ou r_i que tenha a maior largura. No caso das peças p_i e r_i apresentarem a mesma medida na largura, uma

destas é selecionada aleatoriamente. Nesse processo de melhoria, determina-se, também, o número de peças deste tipo a serem incluídas na faixa F_H . Sendo assim, esta quantidade de peças está diretamente relacionada ao comprimento da faixa guilhotina e a sua demanda d_i . A inclusão desta peça, na maior quantidade possível, implicará em uma atualização do conjunto B_H , com a finalidade de se promover um melhor aproveitamento da faixa F_H .

Análogo ao B_H , o conjunto B_V é formado pelas peças de C' , que ao serem incluídas a faixa F_V , não inviabilizaria a sua execução. Entre as peças deste conjunto, uma peça p_i ou r_i é incluída na faixa F_V desde que esta peça apresente o maior comprimento. A quantidade de peças a serem introduzidas na faixa F_V é determinada de acordo com a largura da faixa guilhotina e a sua demanda d_i . O conjunto B_V , assim como o B_H , também é atualizado, até não existir mais uma peça que possa melhorar a faixa.

Ao término desta fase, a faixa guilhotina que apresentar o menor percentual de perda interna será incluída no padrão de corte. Caso haja empate nos valores de perda apresentados pelas faixas F_H e F_V , a escolha da faixa a compor o padrão de corte é aleatória.

5.2 ALGORITMO GRASP-2D

Nesta seção apresentar-se-á um algoritmo baseado na metodologia GRASP para solução do problema de corte bidimensional guilhotinado e restrito, objeto do presente estudo.

Neste espaço pretende-se, em primeiro plano, mostrar as variáveis envolvidas no algoritmo GRASP-2D, bem como o seu pseudocódigo. Em relação às variáveis tem-se:

- m : número de peças distintas;
- p_i : peça i a ser produzida, $i = 1, \dots, m$;
- c_i : comprimento da peça i , $i = 1, \dots, m$;

- l_i : largura da peça i , $i = 1, \dots, m$;
- d_i : limite máximo de peças i a ser produzida, $i = 1, \dots, m$;
- v_i : valor de utilidade da peça i , $i = 1, \dots, m$;
- R : placa retangular em estoque;
- α : parâmetro que controla o grau de miopia e aleatoriedade da fase de construção;
- R_F : sub-retângulo resultante de um corte guilhotinado que define uma faixa em R ;
- $maxiter$: número máximo de iterações estipulado como critério de parada;
- S^* : melhor solução encontrada;
- $aprovS^*$: valor percentual de aproveitamento da melhor solução;
- S : solução corrente;
- $aprovS$: valor percentual de aproveitamento da solução corrente;
- C : lista das peças p_i posicionadas em ordem decrescente, por área;
- C' : lista das próximas peças candidatas a compor uma solução;
- $padcorte_{(g)}$: conjunto das faixas guilhotina que definem um padrão de corte;
- g : número de padrões encontrados em uma solução;
- p_{menor} : peça p_i com menor área e demanda d_i não atendida;
- β : peça p_i com maior área encontrada na lista C' ;
- LRC : conjunto restrito composto pelos melhores elementos da lista C' ;
- F_H : faixa guilhotina horizontal;
- MF_H : sub-retângulo de uma faixa guilhotina horizontal;
- $perdaintF_H$: valor da perda interna em uma faixa guilhotina horizontal;
- F_V : faixa guilhotina vertical;
- MF_V : sub-retângulo de uma faixa guilhotina vertical;
- $perdaintF_V$: valor da perda interna em uma faixa guilhotina vertical;
- B_H : conjunto das próximas peças candidatas a efetuar uma melhoria na faixa F_H ;
- B_V : conjunto das próximas peças candidatas a efetuar uma melhoria na faixa F_V ;
- F_{melhor} : faixa guilhotina incluída em um padrão de corte;
- $perdaintF_{melhor}$: valor percentual da perda interna em uma faixa guilhotina.

Em seguida, descreve-se o pseudocódigo do algoritmo GRASP-2D.

1. Entrar com os dados $(p_i, d_i, R, \alpha, maxiter)$.
2. Fazer $S^* = \emptyset$ e $aprovS^* = 0$.
3. Se o número de iterações é maior que $maxiter$, ir para passo 52.
4. Fazer $S = \emptyset$.
5. Definir a lista $C = (p_1, \dots, p_m)$.
6. Fazer $g = 1$.
7. Se $|C| = 0$ uma solução foi gerada, ir para o passo 49.
8. Fazer $padcorte_{(g)} = \emptyset$.
9. Considerar $R_F = R$, a perda externa inicial de um padrão de corte.
10. Criar $C' = (p_1, \dots, p_t)$, $t \leq m$.
11. Identificar a peça com menor área (p_{menor}) na lista C' ,

$$p_{menor} = \min \{p_i; p_i \in C'\}$$

12. Se $|C'| = 0$ ou p_{menor} não puder ser obtida em R_F , ir para o passo 46.

Fase de Construção

13. Identificar a peça com maior área (β) na lista C' ,

$$\beta = \max \{v(p_i); p_i \in C'\}$$

14. Criar o conjunto LRC ,

$$LRC = \{p_i \in C' \mid \alpha \cdot \beta \leq v(p_i) \leq \beta\}$$

15. Escolher, aleatoriamente, uma peça (p_k) de LRC .
16. Se a peça p_k não puder ser obtida em $R_F = (C_F, L_F)$, excluir p_k da lista C' e voltar ao passo 10.
17. Calcular o número (δ) de peças $p_k = (a, b)$ incluídas na faixa $F_H = (C_F, b)$,

$$\delta = \min \{d_k, \lfloor C_F/a \rfloor\}$$

18. Determinar o retângulo ($MF_H = (X_H, Y_H)$) a ser explorado na fase de melhoria,

$$MF_H = (C_F - \delta a, b)$$

19. Considerar $perdaintF_H = v(MF_H)$ a perda interna inicial de F_H .
20. Calcular o número (ε) de peças $p_k = (a, b)$ incluídas na faixa $F_V = (a, L_F)$,

$$\varepsilon = \min \{d_k, \lfloor L_F/b \rfloor\}$$

21. Determinar o retângulo ($MF_V = (X_V, Y_V)$) a ser explorado na fase de melhoria,

$$MF_V = (a, L_F - \varepsilon b)$$

22. Considerar $perdaintF_V = v(MF_V)$ a perda interna inicial de F_V .

Fase de Melhoria

23. Criar um conjunto B_H com as peças que possam ser incluídas em MF_H ,

$$B_H = \{p_i \in C' / (c_i \leq X_H \text{ e } l_i \leq Y_H) \text{ ou } (l_i \leq X_H \text{ e } c_i \leq Y_H)\}$$

24. Se $|B_H| = 0$, ir para o passo 31.

25. Identificar em B_H , a peça ($p_b = (r, s)$) mais interessante a ser utilizada;

$$s = \max \{l_i, p_i \in B_H\} \text{ ou } s = \max \{c_i, p_i \in B_H\}$$

26. Calcular o número (δ') de peças $p_b = (r, s)$ incluídas na faixa guilhotina F_H ,

$$\delta' = \min \{d_b, \lfloor X_H/r \rfloor\}$$

27. Atualizar o retângulo MF_H ,

$$MF_H = (X_H - \delta'r, Y_H)$$

28. Atualizar a perda interna inicial de F_H ,

$$perdaintF_H = perdaintF_H - \delta'v(p_b)$$

29. Atualizar o conjunto B_H .

30. Voltar ao passo 24.

31. Criar um conjunto B_V com as peças que possam ser incluídas em MF_V ,

$$B_V = \{p_i \in C' / (c_i \leq X_V \text{ e } l_i \leq Y_V) \text{ ou } (l_i \leq X_V \text{ e } c_i \leq Y_V)\}$$

32. Se $|B_V| = 0$, ir para o passo 40.

33. Identificar em B_V , a peça ($p_b = (r, s)$) mais interessante a ser utilizada;

$$r = \max \{c_i, p_i \in B_V\} \text{ ou } r = \max \{l_i, p_i \in B_V\}$$

34. Calcular o número (ε') de peças $p_b = (r, s)$ incluídas na faixa guilhotina F_V ,

$$\varepsilon' = \min \{d_b, \lfloor Y_H/s \rfloor\}$$

35. Atualizar o retângulo MF_V ,

$$MF_V = (X_V, Y_V - \varepsilon' s)$$

36. Atualizar a perda interna inicial de F_V ,

$$perdaintF_V = perdaintF_V - \varepsilon' v(p_b)$$

38. Atualizar o conjunto B_V .

39. Voltar ao passo 32.

40. Determinar a faixa guilhotina com menor percentual de perda,

$$perdaintF_{melhor} = \min \{perdaintF_H/v(F_H), perdaintF_V/v(F_V)\}$$

41. Incluir a faixa guilhotina (F_{melhor}), associada à $perdaintF_{melhor}$, no padrão de corte,

$$padcorte_{(g)} = padcorte_{(g)} \cup \{F_{melhor}\}$$

42. Atualizar o retângulo R_F ,

$$R_F = R_F - F_{melhor}$$

43. Atualizar a demanda d_i das peças p_i incluídas em F_{melhor} .

44. Se $d_i = 0$, excluir peça p_i da lista C .

45. Voltar ao passo 10.

46. Incluir o padrão de corte na solução (S) processada,

$$S = S \cup padcorte_{(g)}$$

47. Incrementar a variável g ,

$$g = g + 1$$

48. Voltar ao passo 7.

49. Calcular o valor percentual do aproveitamento ($aprovS$) associada a solução S ,

$$aprovS = \frac{\sum v(p_i)}{g \cdot v(R)}$$

50. Se $aprovS > aprovS^*$, atualizar S^* e $aprovS^*$.

$$\begin{cases} S^* = S; \\ aprovS^* = aprovS \end{cases}$$

51. Voltar ao passo 3.

52 Escrever $(S^*, aprovS^*)$.

53 Fim.

6. RESULTADOS COMPUTACIONAIS

Neste capítulo são conferidos os resultados computacionais obtidos com o algoritmo proposto GRASP-2D apresentado na seção 5.2. Este algoritmo foi implementado em linguagem C, utilizando o compilador Microsoft Visual C++ 6.0. Os testes foram realizados em um microcomputador Pentium 3, com processador Intel de 801Mhz e 192 MB de memória RAM.

Os testes com o algoritmo GRASP-2D foram realizados com um grupo de 24 instâncias encontradas em Amaral (1994). Em seu trabalho, Amaral considera o problema de gerar padrões de corte bidimensionais guilhotinados e restritos de uma forma ótima. Para solução deste problema, duas implementações são propostas, uma delas está fundamentada na técnica Busca Tabu e a outra na técnica *Simulated Annealing*. A performance do algoritmo Busca Tabu foi testada com execuções partindo de uma solução inicial construída ou de uma solução aleatória. Já o desempenho do *Simulated Annealing*, foi avaliado somente com soluções iniciais construídas. Esta abordagem é considerada pioneira no que diz respeito à utilização das metodologias Busca Tabu e *Simulated Annealing* para o tratamento do problema de corte bidimensional.

A fim de validar o algoritmo proposto, são comparados os valores da função objetivo fornecido pelos algoritmos Busca Tabu e o *Simulated Annealing* com os valores apresentados com algoritmo GRASP-2D quando aplicados as mesmas instâncias. É importante ressaltar que os valores de função objetivo, apresentados em Amaral (1994), informam o percentual de preenchimento da placa obtido pelo melhor padrão de corte. Logo, os resultados que serão confrontados nesta dissertação referem-se ao percentual de aproveitamento do melhor padrão de corte, gerado pelo GRASP-2D para cada instância, respectivamente.

As vinte e quatro instâncias utilizadas nesta dissertação estão divididas em duas classes: a primeira com instâncias encontradas na literatura e a segunda com instâncias geradas aleatoriamente. Com o parâmetro α assumindo os valores

0.1, 0.25, 0.5, 0.75 e 1.0, chega-se ao total de 120 execuções. Como critério de parada, fica estabelecido o número máximo de 1000 iterações. Assim, a cada instância executada pelo algoritmo GRASP-2D é registrada a solução com maior percentual de aproveitamento, em um total de 5000 soluções geradas. Algumas dessas soluções, geradas nos testes computacionais com o algoritmo GRASP-2D, utilizando instâncias clássicas da literatura, serão exibidas no Apêndice.

A seguir são apresentados os resultados encontrados para as respectivas classes de instâncias.

6.1 RESULTADOS PARA A PRIMEIRA CLASSE DE INSTÂNCIAS

Na primeira classe há seis instâncias extraídas da literatura. Estes exemplos, referenciados no quadro 6.1, são constantemente utilizados em comparações de métodos para o problema de corte bidimensional guilhotinado e restrito. Os dados desta classe de instâncias encontram-se no Anexo 1.

Quadro 6.1 - Referências bibliográficas da primeira classe de instâncias.

Código	Itens	Objeto	Referências Bibliográficas
L1	7	(15,10)	Christofides e Whitlock (1977)
L2	5	(60,108)	Wang(1983)
L3	10	(40,70)	Oliveira e Ferreira(1990)
L4	10	(40,70)	Oliveira e Ferreira(1990)
L5	10	(40,70)	Christofides e Whitlock (1977)
L6	20	(40,70)	Christofides e Whitlock (1977), Wang(1983)

A tabela 6.1 apresenta os resultados obtidos com o algoritmo GRASP-2D para esta classe de instâncias, usando o parâmetro α igual a 0.1, 0.25, 0.5, 0.75 e 1.0. O tempo computacional t para cada instância com variação do parâmetro α é medido em segundos.

Tabela 6.1 - Resultados da primeira classe de instâncias com o GRASP-2D.

Código	$\alpha = 0.1$	t(s)	$\alpha = 0.25$	t(s)	$\alpha = 0.5$	t(s)	$\alpha = 0.75$	t(s)	$\alpha = 1.0$	t(s)
L1	100.00	27	100.00	27	100.00	27	100.00	27	100.00	30
L2	100.00	42	100.00	42	100.00	47	100.00	48	100.00	48
L3	96.89	36	97.32	38	96.89	38	96.61	37	90.29	39
L4	95.57	39	96.93	38	95.86	40	95.86	38	92.86	38
L5	97.25	36	95.57	37	95.11	39	94.43	38	94.43	36
L6	98.36	79	98.36	81	98.36	81	98.36	84	97.21	84

Os resultados apresentados pelos algoritmos Busca Tabu, *Simulated Annealing* e GRASP-2D, para as instâncias desta classe, são conferidos na tabela 6.2. Evidencia-se, nesta tabela, que o algoritmo Busca Tabu foi executado a partir de soluções iniciais construídas e aleatórias. Já na execução do algoritmo *Simulated Annealing*, o ponto de partida foi uma solução construída.

Tabela 6.2 - Resultados obtidos da primeira classe de instâncias.

Código	Aproveitamento da Placa (%)			
	Busca Tabu (BT*) (Sol. inicial aleatória)	Busca Tabu (BT) (Sol. Inicial não-aleatória)	<i>Simulated Annealing</i> (Sol. Inicial não-aleatória)	GRASP-2D
L1	96.00	98.66	96.00	100.00*
L2	97.28	97.16	96.30	100.00*
L3	95.43	96.52	95.43	97.32*
L4	95.00	95.00	93.00	96.93*
L5	96.21	95.89	93.00	97.25*
L6	97.98	97.40	96.43	98.36*

* Melhor valor da função objetivo

Analisando os resultados mostrados pela tabela 6.2, verifica-se que em nenhum caso o Busca Tabu e o *Simulated Annealing* apresentou resultados melhores do que os obtidos com o GRASP-2D. A superioridade do algoritmo proposto é evidenciada observando a diferença entre os valores de função objetivo apresentado para algumas instâncias dessa classe. Para instância L3, o Busca Tabu

obteve o pior resultado comparado ao GRASP-2D. Já com o *Simulated Annealing*, a diferença entre os valores chegou a 4.25% para instância L5.

Baseado nesta tabela, a figura 6.1 apresentará um comparativo entre os valores obtidos para função objetivo nas seis instâncias executadas.

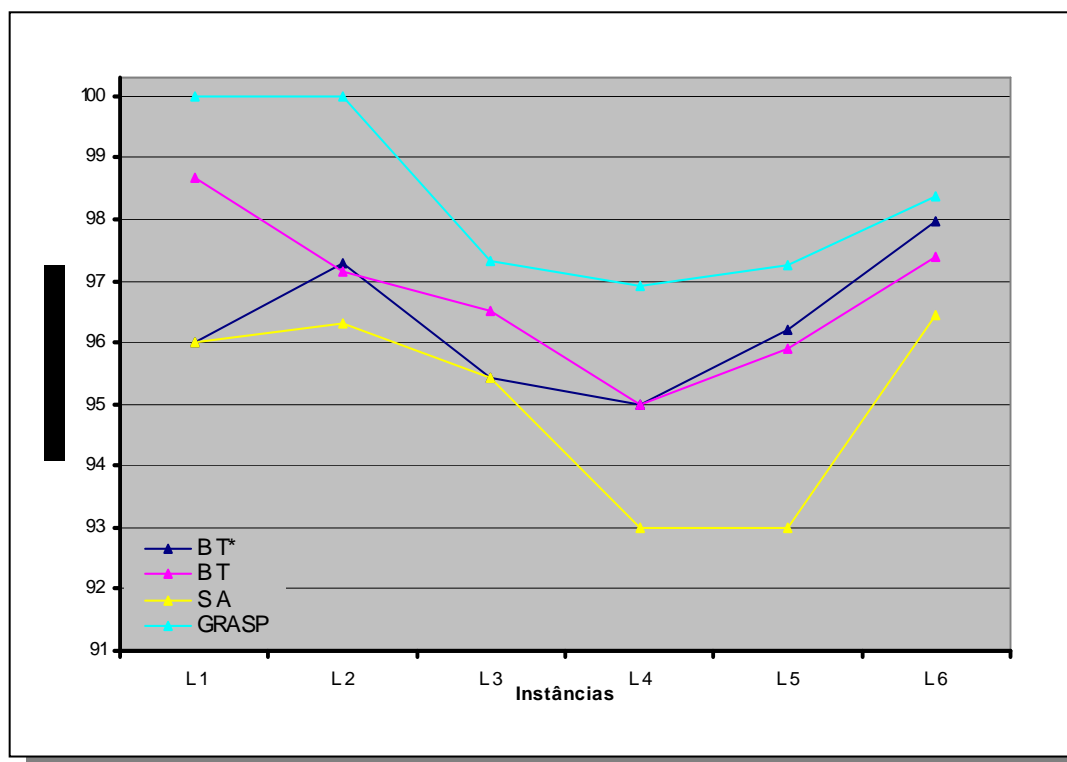


Figura 6.1 - Comparativo dos resultados executando os algoritmos para as instâncias da primeira classe.

Os resultados obtidos com a utilização do algoritmo proposto, em instâncias desta classe, foram bastante satisfatórios. Observe na figura 6.1 que, para todas as instâncias, os resultados encontrados com o GRASP-2D foram superiores aos obtidos com os algoritmos *Simulated Annealing* e Busca Tabu.

As figuras a seguir apresentarão os padrões de corte encontrados pelo GRASP-2D nas respectivas instâncias.

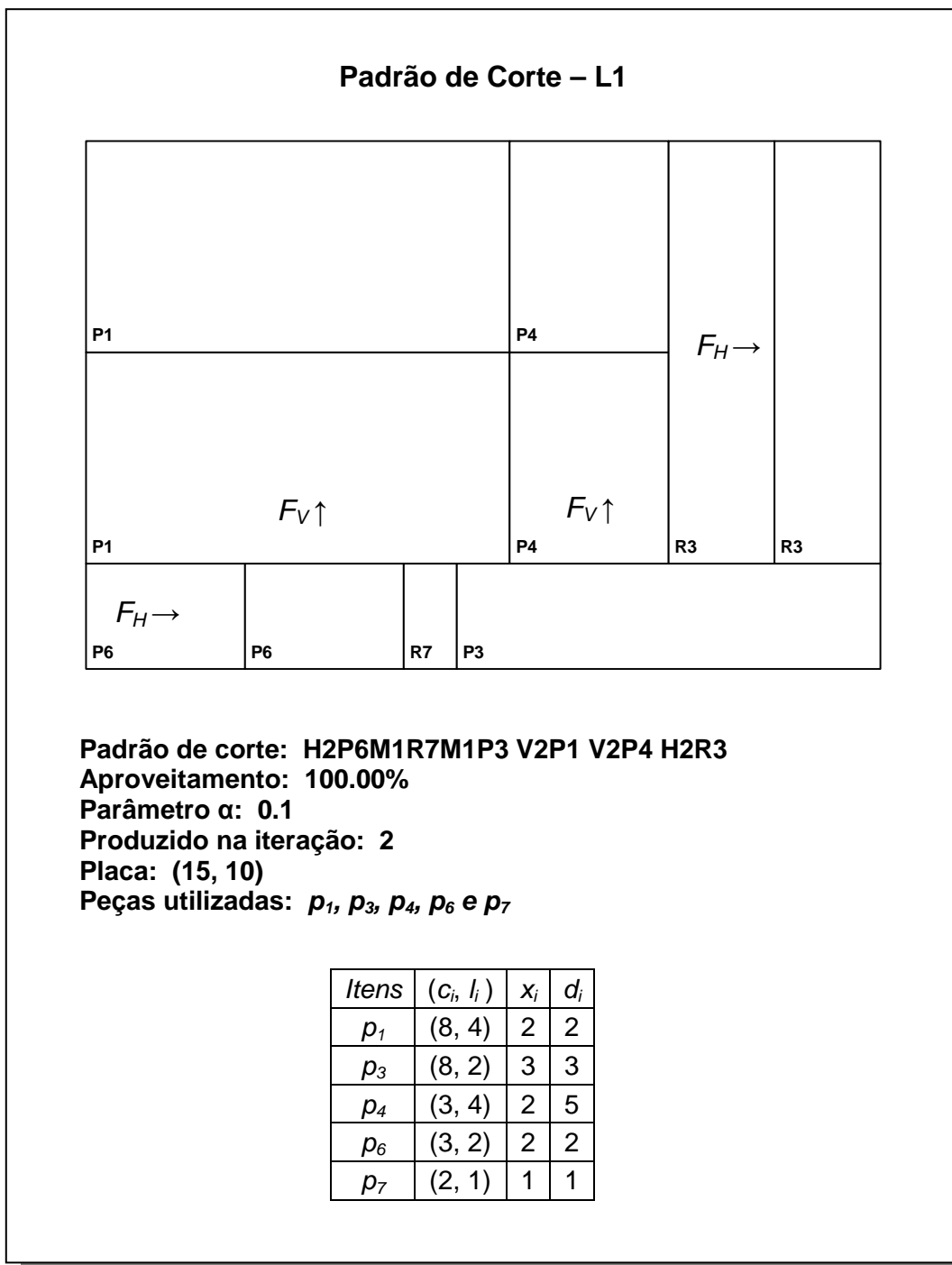


Figura 6.2 - Padrão de corte produzido para L1.

Padrão de Corte – L2

$F_H \rightarrow$		
P2	P2	P2
$F_H \rightarrow$		
R1	R1	
$F_H \rightarrow$		
R1	R1	
$F_H \rightarrow$		
R1	R1	

Padrão de corte: H2R1 H2R1 H2R1 H3P2

Aproveitamento: 100.00%

Parâmetro α : 0.75

Produzido na iteração: 16

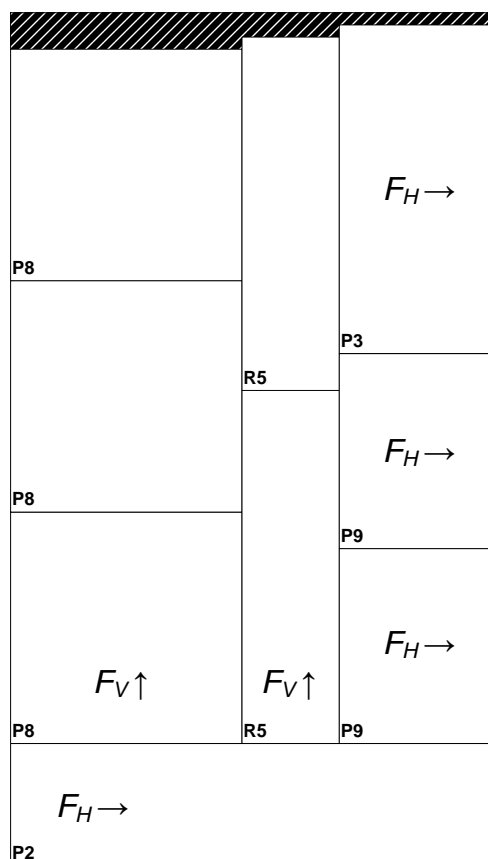
Placa: (60, 108)

Peças utilizadas: p_1 e p_2

<i>Itens</i>	(c_i, l_i)	x_i	d_i
p_1	(28, 30)	6	9
p_2	(20, 24)	3	9

Figura 6.3 - Padrão de corte produzido para L2.

Padrão de Corte – L4



Padrão de corte: H1P2 V3P8 V2R5 H1P9 H1P9 H1P3

Aproveitamento: 96.93%

Parâmetro α : 0.25

Produzido na iteração: 18

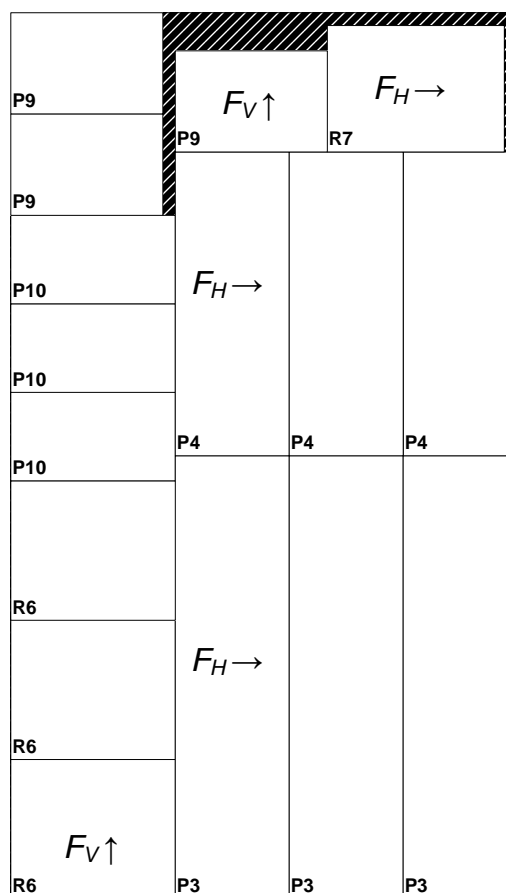
Placa: (40, 70)

Peças utilizadas: p_2 , p_3 , p_5 , p_8 e p_9

<i>Itens</i>	(c_i, l_i)	x_i	d_i
p_2	(40, 10)	1	1
p_3	(13, 27)	1	3
p_5	(29, 8)	2	4
p_8	(19, 19)	3	4
p_9	(13, 16)	2	2

Figura 6.5 - Padrão de corte produzido para L4.

Padrão de Corte – L5



Padrão de corte: V3R6M3P10M2P9 H3P3 H3P4 V1P9 H1R7

Aproveitamento: 97.25%

Parâmetro α : 0.1

Produzido na iteração: 922

Placa: (40, 70)

Peças utilizadas: p_3, p_4, p_6, p_7, p_9 e p_{10}

Itens	(c_i, l_i)	x_i	d_i
p_3	(9, 35)	3	3
p_4	(9, 24)	3	3
p_6	(11, 13)	3	3
p_7	(10, 14)	7	1
p_9	(12, 8)	3	3
p_{10}	(13, 7)	3	3

Figura 6.7 - Padrão de corte produzido para L5.

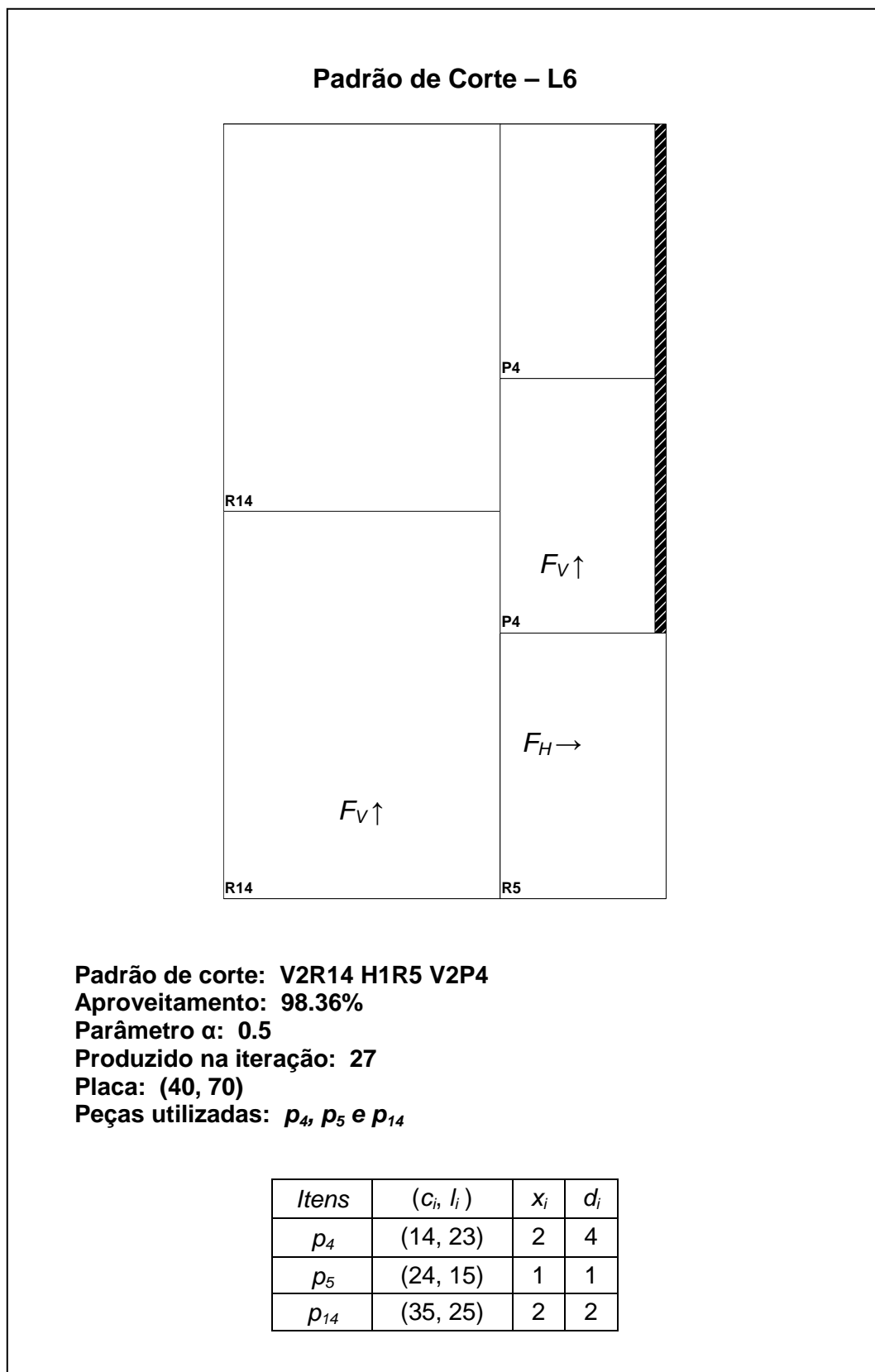


Figura 6.8 - Padrão de corte produzido para L6.

6.2 RESULTADOS PARA A SEGUNDA CLASSE DE INSTÂNCIAS

A segunda classe é constituída de dezoito instâncias originadas a partir de modificações nas instâncias apresentadas por Wang (1983), Oliveira e Ferreira (1990), ou ainda, adaptações nas três instâncias apresentadas por Christofides e Whitlock (1977). É importante destacar que esta classe de problemas-testes é constituída originalmente de vinte problemas, porém os problemas dezessete e dezoito não apresentam restrições de demanda. As instâncias consideradas são encontradas no Anexo 2.

A seguir será mostrada uma tabela com os valores obtidos para função objetivo nas dezoito instâncias executadas com o algoritmo GRASP-2D. Novamente, o tempo de execução é t calculado em segundos.

Tabela 6.3 - Resultados da segunda classe de instâncias com o GRASP-2D.

Código	$\alpha = 0.1$	t(s)	$\alpha = 0.25$	t(s)	$\alpha = 0.5$	t(s)	$\alpha = 0.75$	t(s)	$\alpha = 1.0$	t(s)
A1	100.00	38	100.00	40	100.00	41	100.00	41	100.00	42
A2	100.00	39	100.00	41	97.27	44	90.91	43	90.91	43
A3	99.46	51	99.69	56	97.90	57	96.51	65	95.11	67
A4	100.00	55	100.00	57	100.00	61	100.00	63	100.00	61
A5	100.00	37	100.00	39	100.00	40	100.00	42	100.00	41
A6	100.00	46	100.00	47	100.00	48	99.13	47	98.04	48
A7	100.00	49	100.00	53	100.00	57	100.00	58	100.00	60
A8	100.00	45	100.00	48	100.00	51	100.00	51	100.00	54
A9	99.61	39	99.61	39	98.83	45	99.22	46	98.05	45
A10	100.00	49	100.00	51	100.00	60	100.00	61	98.21	62
A11	100.00	44	100.00	46	100.00	49	100.00	48	100.00	45
A12	100.00	44	100.00	45	100.00	47	100.00	49	100.00	50
A13	100.00	54	100.00	59	100.00	64	100.00	66	100.00	68
A14	100.00	57	100.00	63	100.00	69	100.00	68	100.00	69
A15	100.00	31	100.00	31	100.00	31	100.00	32	100.00	33
A16	98.67	44	98.31	46	97.07	47	96.44	54	96.44	54
A17	100.00	48	100.00	49	100.00	53	100.00	53	97.54	53
A18	99.23	44	99.82	49	99.44	54	99.27	52	95.91	53

Na próxima tabela serão apresentados os valores obtidos pelos algoritmos *Simulated Annealing*, Busca Tabu e GRASP-2D para as instâncias desta classe. Destaca-se, nesta tabela, que o algoritmo *Simulated Annealing* foi executado somente a partir de uma solução inicial construída, enquanto o algoritmo Busca Tabu teve como ponto de partida soluções construídas e aleatórias.

Tabela 6.4 - Resultados obtidos da segunda classe de instâncias.

Código	Aproveitamento da Placa (%)			
	Busca Tabu (BT*) (Sol. inicial aleatória)	Busca Tabu (BT) (Sol. Inicial não-aleatória)	<i>Simulated Annealing</i> (Sol. Inicial não-aleatória)	GRASP-2D
A1	100.00*	100.00*	99.00	100.00*
A2	96.82	100.00*	98.81	100.00*
A3	98.00	98.00	96.90	99.69*
A4	100.00*	100.00*	98.13	100.00*
A5	97.98	100.00*	96.90	100.00*
A6	99.13	100.00*	100.00*	100.00*
A7	98.00	98.00	97.00	100.00*
A8	98.32	98.32	97.76	100.00*
A9	99.61*	99.61*	95.97	99.61*
A10	100.00*	100.00*	97.70	100.00*
A11	98.91	100.00*	96.20	100.00*
A12	98.79	100.00*	95.16	100.00*
A13	97.40	97.40	96.20	100.00*
A14	96.10	96.10	95.00	100.00*
A15	99.21	100.00*	95.24	100.00*
A16	95.59	96.72	92.10	98.67*
A17	96.62	98.61	93.53	100.00*
A18	93.43	94.51	91.43	99.82*

* Melhor valor da função objetivo

Com base nesta tabela, a figura 6.8 apresentará um comparativo entre os melhores valores percentuais obtidos em um padrão de corte, considerando os procedimentos executados.

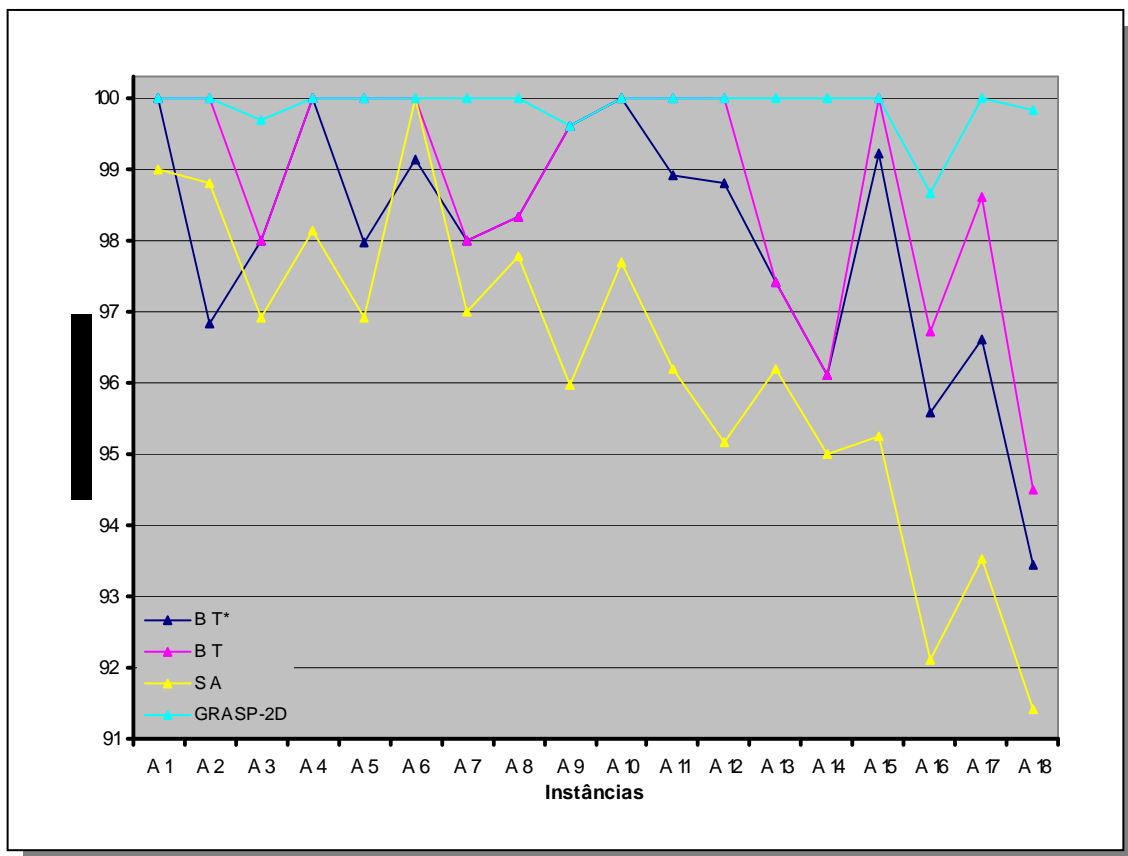
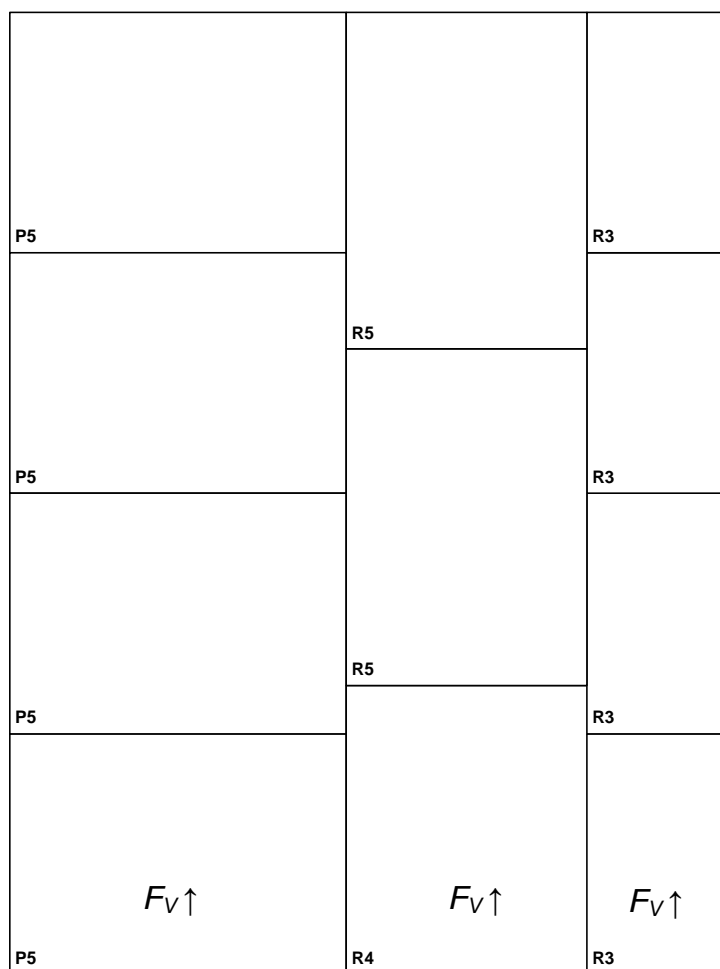


Figura 6.8 - Comparativo dos resultados para as instâncias da segunda classe.

Como pode ser observado, todos os padrões de corte gerados pelo GRASP-2D, para essa classe de instâncias, apresentam valores de função objetivo iguais ou superiores aos obtidos pelos outros procedimentos. Das dezoito instâncias analisadas, tanto o *Simulated Annealing* quanto o Busca Tabu não produziram soluções superiores as encontradas pelo GRASP-2D. Em dez instâncias dessa classe, o Busca Tabu produziu soluções com a mesma qualidade das encontradas pelo GRASP-2D. Já o *Simulated Annealing*, obteve apenas uma solução com rendimento idêntico ao obtido pelo GRASP-2D. Em algumas instâncias o algoritmo proposto mostrou-se bastante eficiente, superando em mais de 5% os valores obtidos pelos outros algoritmos.

Em seqüência, as figuras apresentarão os melhores padrões de corte obtidos com a execução do GRASP-2D nas respectivas instâncias dessa classe.

Padrão de Corte – A1



Padrão de corte: V4P5 V1R4M2R5 V4R3

Aproveitamento: 100.00%

Parâmetro α : 0.5

Produzido na iteração: 22

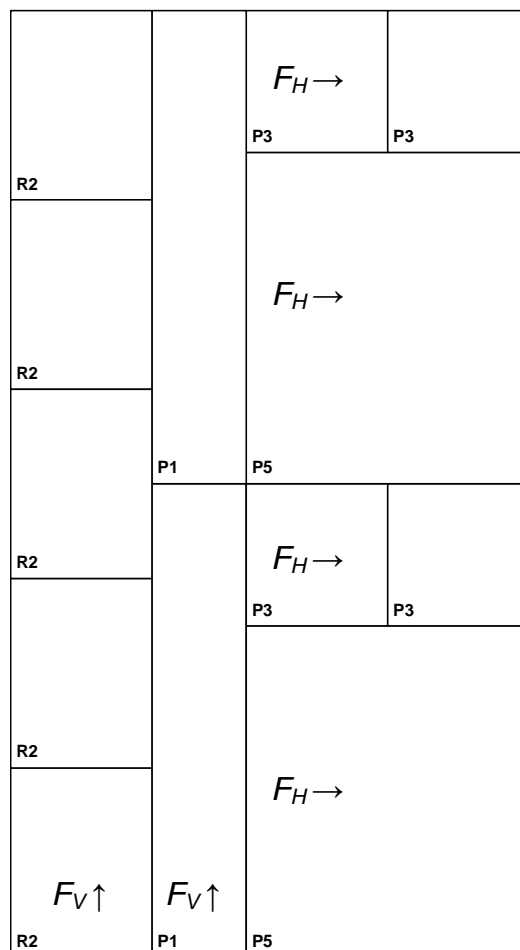
Placa: (15, 20)

Peças utilizadas: p_3 , p_4 e p_5

<i>Itens</i>	(c_i, l_i)	x_i	d_i
p_3	(5, 3)	4	8
p_4	(6, 5)	1	8
p_5	(7, 5)	6	8

Figura 6.9 - Padrão de corte produzido para A1.

Padrão de Corte – A2



Padrão de corte: V5R2 V2P1 H1P5 H2P3 H1P5 H2P3

Aproveitamento: 100.00%

Parâmetro α : 0.1

Produzido na iteração: 27

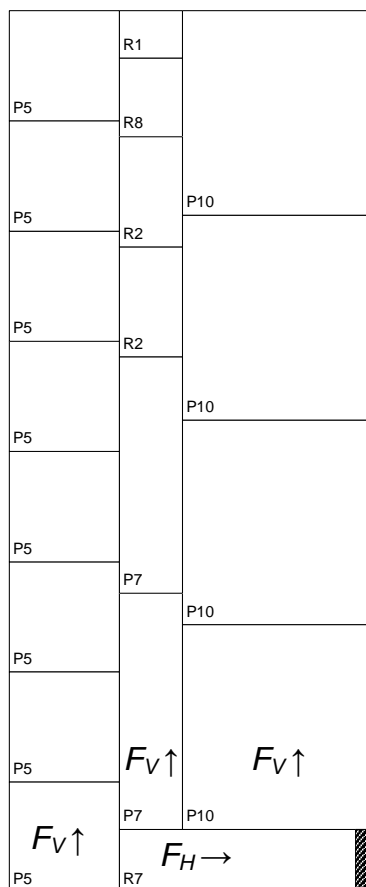
Placa: (11, 20)

Peças utilizadas: p_1 , p_2 , p_3 e p_5

<i>Itens</i>	(c_i, l_i)	x_i	d_i
p_1	(2, 10)	2	7
p_2	(4, 3)	5	7
p_3	(3, 3)	4	7
p_5	(6, 7)	2	9

Figura 6.10 - Padrão de corte produzido para A2.

Padrão de Corte – A3



Padrão de corte: V8P5 H1R7 V2P7M2R2M1R8M1R1 V4P10

Aproveitamento: 99.69%

Parâmetro α : 0.25

Produzido na iteração: 30

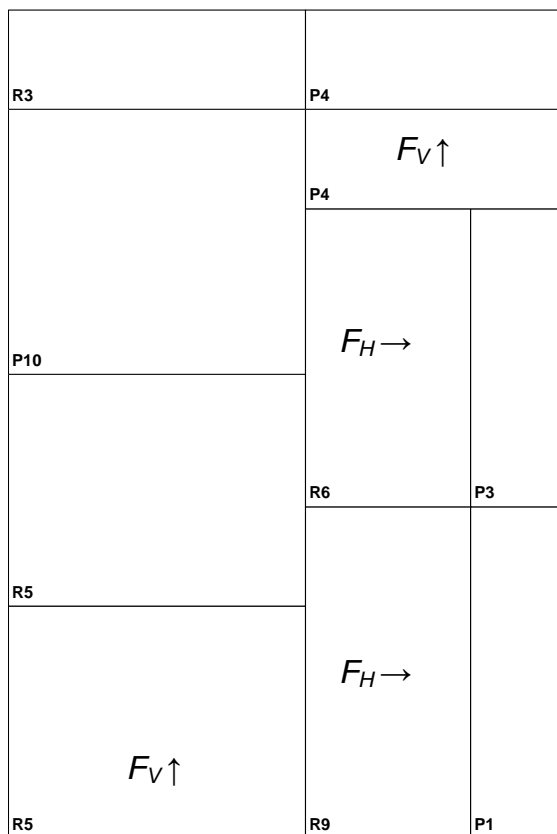
Placa: (23, 56)

Peças utilizadas: p_1, p_2, p_5, p_7, p_8 e p_{10}

Itens	(c_i, l_i)	x_i	d_i
p_1	(3, 4)	1	1
p_2	(7, 4)	2	3
p_5	(7, 7)	8	4
p_7	(4, 15)	3	4
p_8	(5, 4)	1	2
p_{10}	(12, 13)	4	2

Figura 6.11 - Padrão de corte produzido para A3.

Padrão de Corte – A5



Padrão de corte: V2R5M1P10M1R3 H1R9M1P1 H1R6M1P3 V2P4

Aproveitamento: 100.00%

Parâmetro α : 0.75

Produzido na iteração: 1

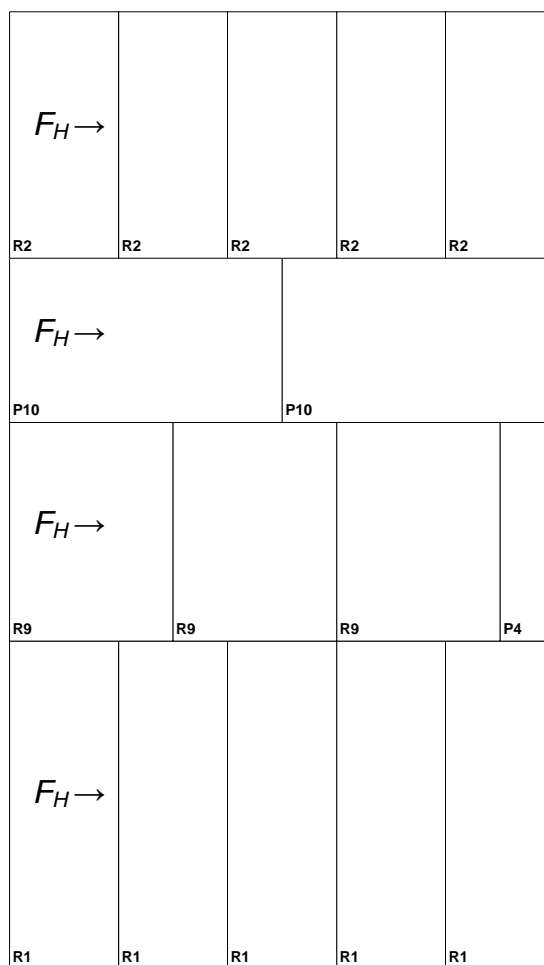
Placa: (17, 25)

Peças utilizadas: $p_1, p_3, p_4, p_5, p_6, p_9$ e p_{10}

Itens	(c_i, l_i)	x_i	d_i
p_1	(3, 10)	1	1
p_3	(3, 9)	2	4
p_4	(8, 3)	2	2
p_5	(7, 9)	2	2
p_6	(9, 5)	1	3
p_9	(10, 5)	1	4
p_{10}	(9, 8)	1	1

Figura 6.13 - Padrão de corte produzido para A5.

Padrão de Corte – A7



Padrão de corte: H5R1 H3R9M1P4 H2P10 H5R2

Aproveitamento: 100.00%

Parâmetro α : 0.5

Produzido na iteração: 17

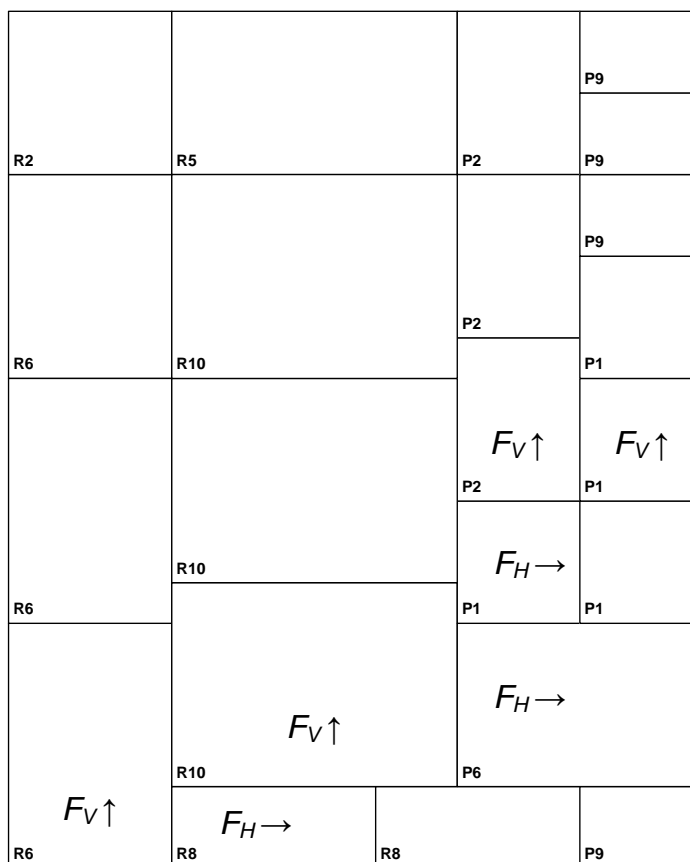
Placa: (20, 35)

Peças utilizadas: p_1, p_2, p_4, p_9 e p_{10}

Itens	(c_i, l_i)	x_i	d_i
p_1	(12, 4)	5	5
p_2	(9, 4)	5	5
p_4	(2, 8)	1	5
p_9	(8, 6)	3	5
p_{10}	(10, 6)	2	5

Figura 6.15 - Padrão de corte produzido para A7.

Padrão de Corte – A8



Padrão de corte: V3R6M1R2 H2R8M1P9 V3R10M1R5 H1P6 H2P1
V3P2 V2P1M3P9

Aproveitamento: 100.00%

Parâmetro α : 0.25

Produzido na iteração: 20

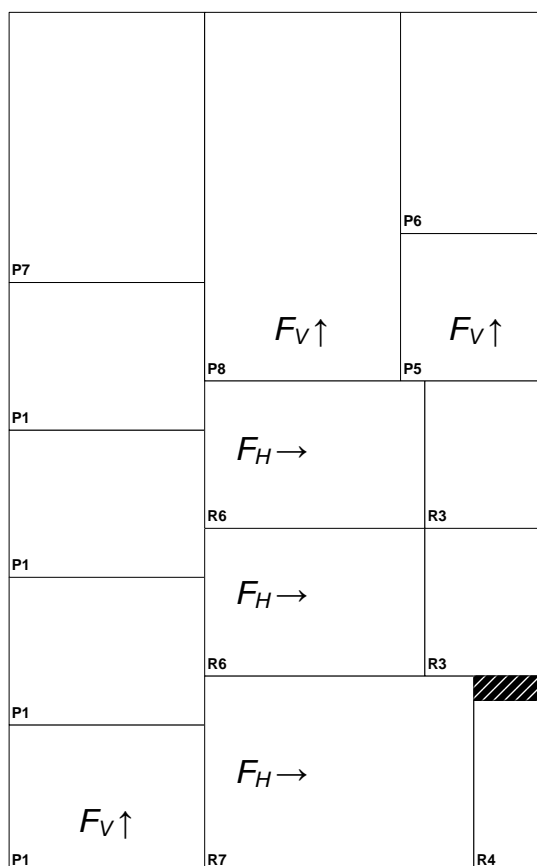
Placa: (17, 21)

Peças utilizadas: $p_1, p_2, p_5, p_6, p_8, p_9$ e p_{10}

Itens	(c_i, l_i)	x_i	d_i
p_1	(3, 3)	4	4
p_2	(3, 4)	4	4
p_5	(4, 7)	1	4
p_6	(6, 4)	4	4
p_8	(2, 5)	2	4
p_9	(3, 2)	4	4
p_{10}	(5, 7)	3	4

Figura 6.16 - Padrão de corte produzido para A8.

Padrão de Corte – A9



Padrão de corte: V4P1M1P7 H1R7M1R4 H1R6M1R3 H1R6M1R3
V1P8 H1P5 H1P6

Aproveitamento: 99.61%

Parâmetro α : 0.25

Produzido na iteração: 90

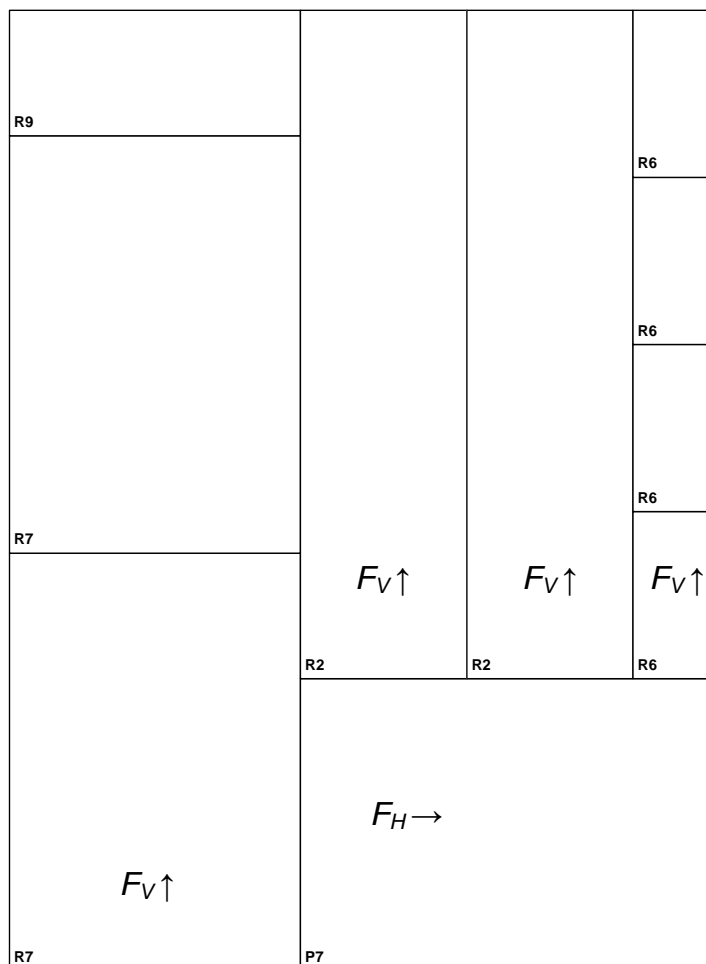
Placa: (22, 35)

Peças utilizadas: $p_1, p_3, p_4, p_5, p_6, p_7$ e p_8

Itens	(c_i, l_i)	x_i	d_i
p_1	(8, 6)	4	4
p_3	(6, 5)	2	4
p_4	(7, 3)	1	4
p_5	(6, 6)	1	4
p_6	(6, 9)	3	4
p_7	(8, 11)	1	4
p_8	(8, 15)	1	4

Figura 6.17 - Padrão de corte produzido para A9.

Padrão de Corte – A10



Padrão de corte: V2R7M1R9 H1P7 V1R2 V1R2 V4R6

Aproveitamento: 100.00%

Parâmetro α : 0.75

Produzido na iteração: 12

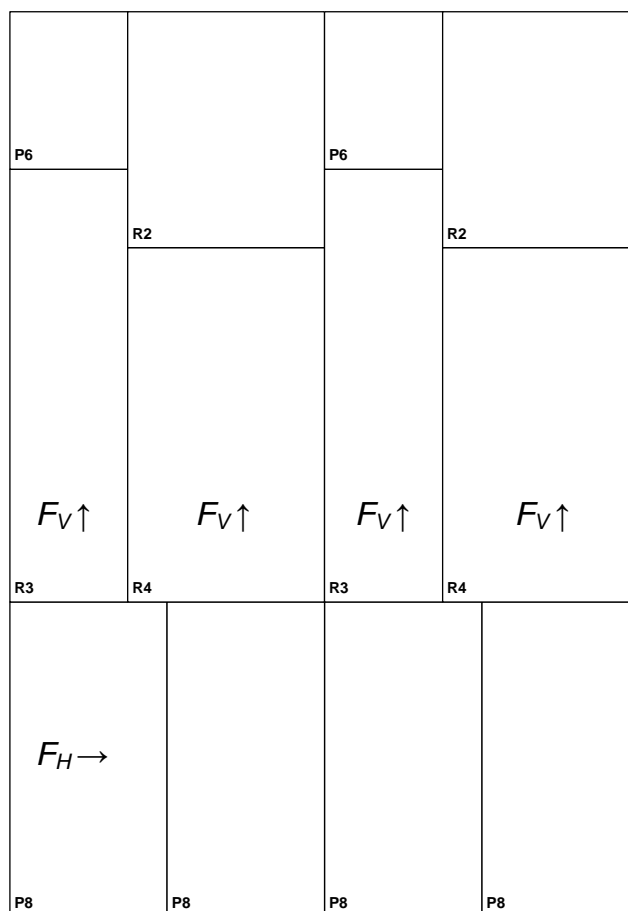
Placa: (17, 23)

Peças utilizadas: p_2 , p_6 , p_7 e p_9

<i>Itens</i>	(C_i, l_i)	x_i	d_i
p_2	(16, 4)	1	1
p_6	(4, 2)	2	4
p_7	(10, 7)	2	2
p_9	(3, 7)	2	2

Figura 6.18 - Padrão de corte produzido para A10.

Padrão de Corte – A11



Padrão de corte: H4P8 V1R3M1P6 V1R4M1R2 V1R3M1P6 V1R4M1R2

Aproveitamento: 100.00%

Parâmetro α : 0.5

Produzido na iteração: 9

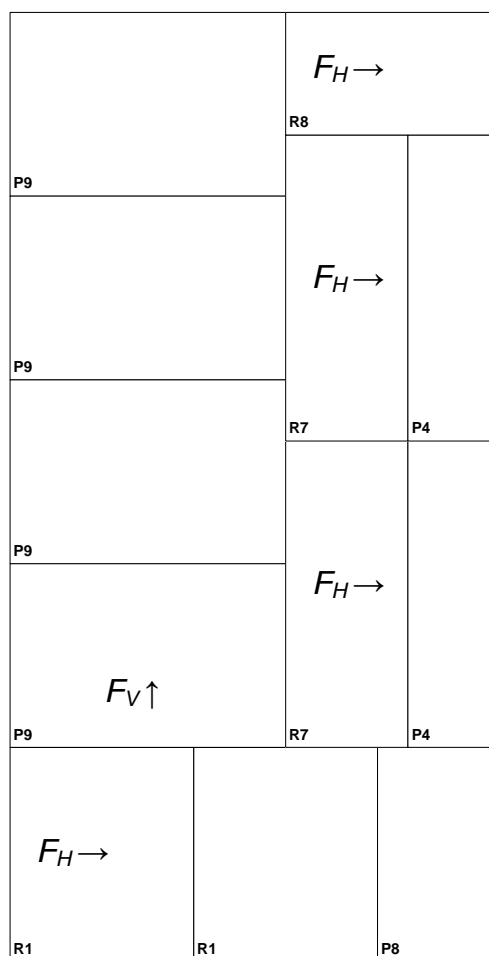
Placa: (16, 23)

Peças utilizadas: p_2, p_3, p_4, p_6 e p_8

<i>Itens</i>	(c_i, l_i)	x_i	d_i
p_2	(6, 5)	2	4
p_3	(11, 3)	2	4
p_4	(9, 5)	2	4
p_6	(3, 4)	2	4
p_8	(4, 8)	4	4

Figura 6.19 - Padrão de corte produzido para A11.

Padrão de Corte – A12



Padrão de corte: H2R1M1P8 V4P9 H1R7M1P4 H1R7M1P4 H1R8

Aproveitamento: 100.00%

Parâmetro α : 0.75

Produzido na iteração: 4

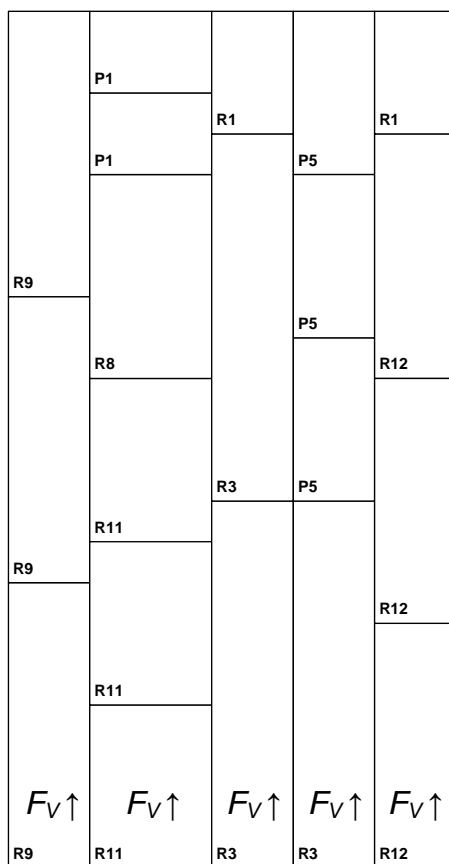
Placa: (16, 31)

Peças utilizadas: p_1, p_4, p_7, p_8 e p_9

<i>Itens</i>	(c_i, l_i)	x_i	d_i
p_1	(7, 6)	2	4
p_4	(3, 10)	2	4
p_7	(10, 4)	2	4
p_8	(4, 7)	2	4
p_9	(9, 6)	4	4

Figura 6.20 - Padrão de corte produzido para A12.

Padrão de Corte – A13



Padrão de corte: V3R9 V3R11M1R8M2P1 V2R3M1R1 V1R3M3P5
V3R12M1R1

Aproveitamento: 100.00%

Parâmetro α : 0.5

Produzido na iteração: 2

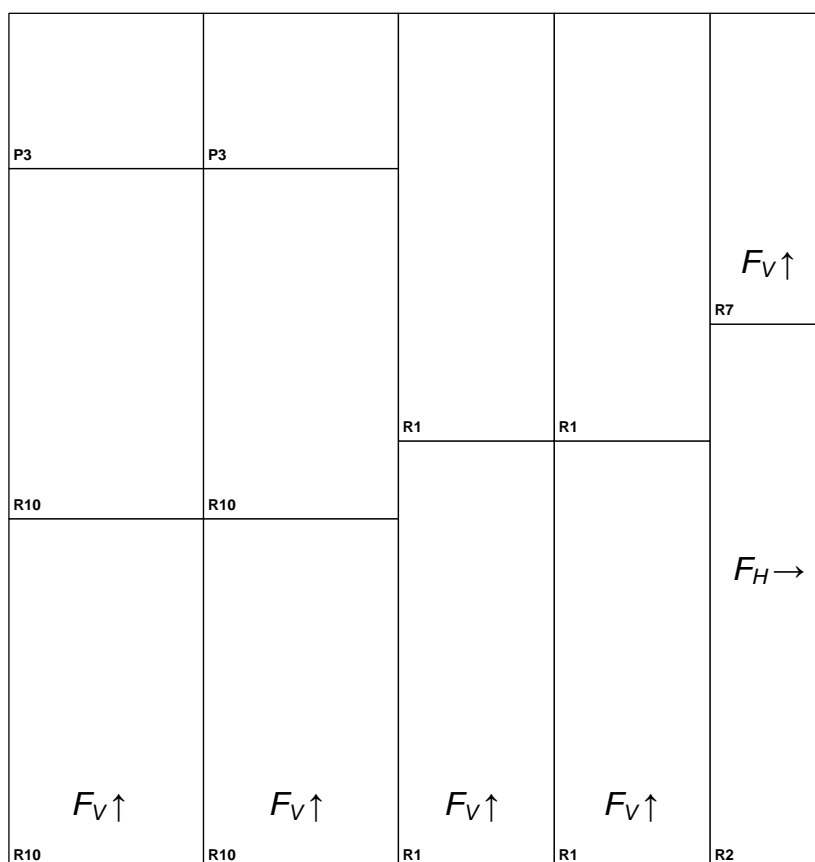
Placa: (11, 21)

Peças utilizadas: $p_1, p_3, p_5, p_8, p_9, p_{11}$ e p_{12}

Itens	(c_i, l_i)	x_i	d_i
p_1	(3, 2)	4	4
p_3	(9, 2)	3	3
p_5	(2, 4)	3	5
p_8	(5, 3)	1	4
p_9	(7, 2)	3	3
p_{11}	(4, 3)	3	3
p_{12}	(6, 2)	3	4

Figura 6.21 - Padrão de corte produzido para A13.

Padrão de Corte – A14



Padrão de corte: V2R10M1P3 V2R10M1P3 V2R1 V2R1 H1R2 V1R7

Aproveitamento: 100.00%

Parâmetro α : 1

Produzido na iteração: 21

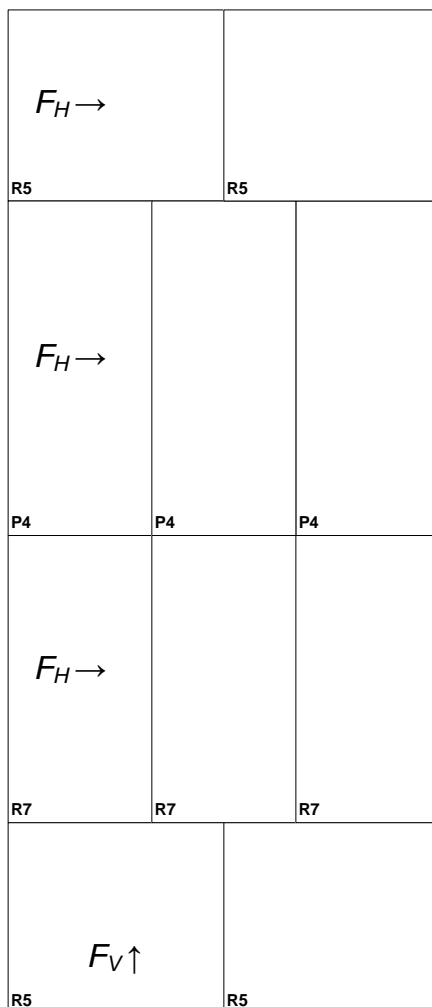
Placa: (21, 22)

Peças utilizadas: p_1, p_2, p_3, p_7 e p_{10}

Itens	(c_i, l_i)	x_i	d_i
p_1	(11, 4)	4	4
p_2	(14, 3)	1	4
p_3	(5, 4)	2	4
p_7	(8, 3)	1	4
p_{10}	(9, 5)	4	4

Figura 6.22 - Padrão de corte produzido para A14.

Padrão de Corte – A15



Padrão de corte: V2R5 H3R7 H3P4 H2R5

Aproveitamento: 100.00%

Parâmetro α : 0.75

Produzido na iteração: 24

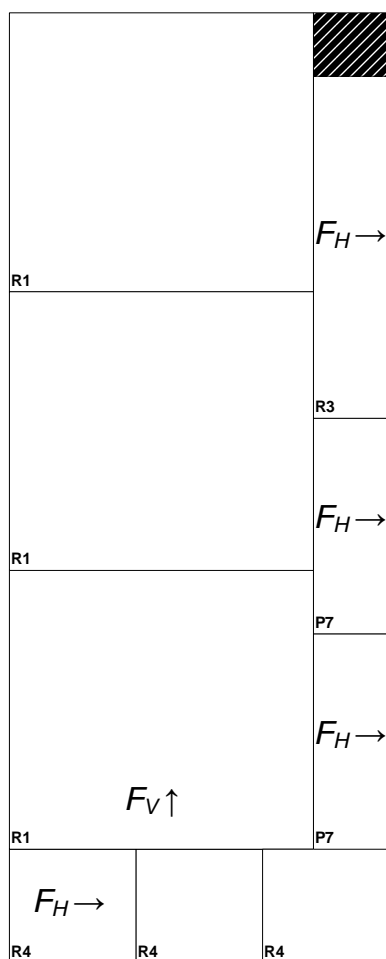
Placa: (18, 42)

Peças utilizadas: p_4 , p_5 e p_7

<i>Itens</i>	(c_i, l_i)	x_i	d_i
p_4	(6, 14)	3	4
p_5	(8, 9)	4	4
p_7	(12, 6)	3	3

Figura 6.23 - Padrão de corte produzido para A15.

Padrão de Corte – A16



Padrão de corte: H3R4 V3R1 H1P7 H1P7 H1R3

Aproveitamento: 98.67%

Parâmetro α : 0.1

Produzido na iteração: 104

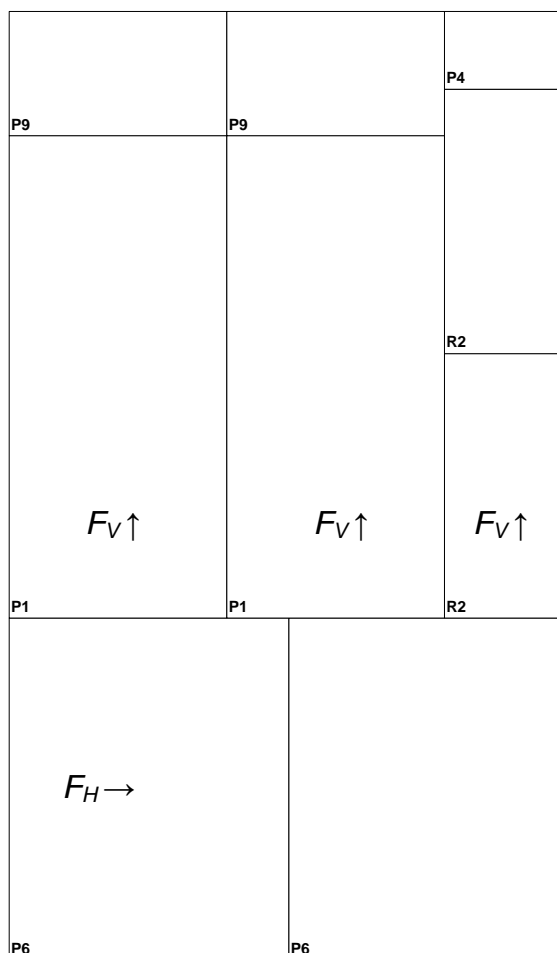
Placa: (30, 75)

Peças utilizadas: p_1 , p_3 , p_4 e p_7

<i>Itens</i>	(c_i, l_i)	x_i	d_i
p_1	(22, 24)	3	4
p_3	(27, 6)	1	4
p_4	(9, 10)	3	4
p_7	(6, 17)	2	3

Figura 6.24 - Padrão de corte produzido para A16.

Padrão de Corte – A17



Padrão de corte: H2P6 V1P1M1P9 V1P1M1P9 V2R2M1P4

Aproveitamento: 100.00%

Parâmetro α : 0.5

Produzido na iteração: 27

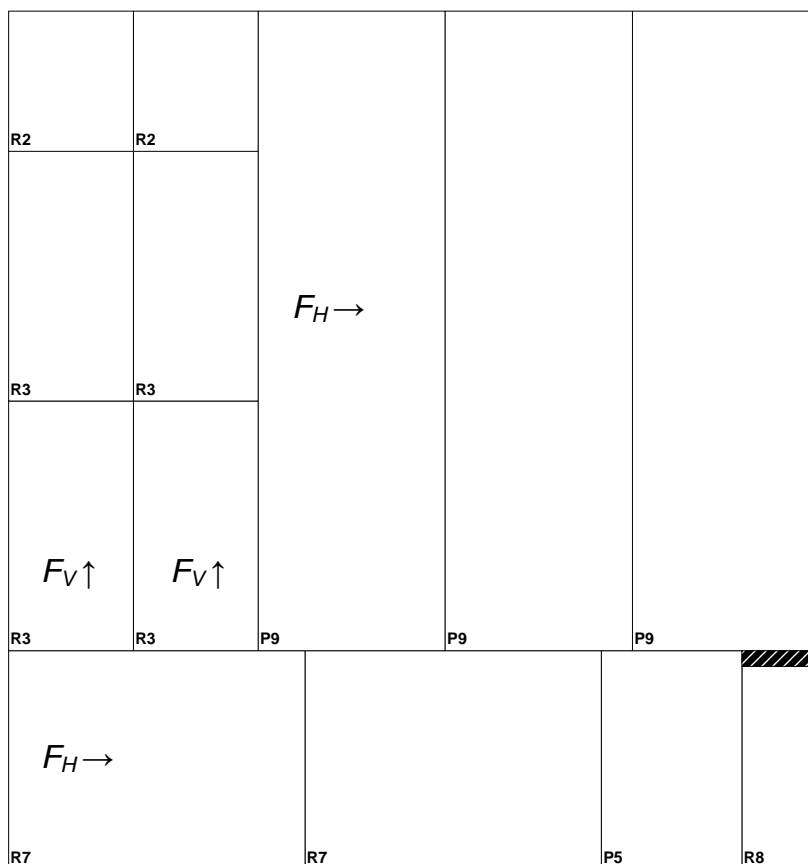
Placa: (36, 61)

Peças utilizadas: p_1, p_2, p_4, p_6 e p_9

<i>Itens</i>	(c_i, l_i)	x_i	d_i
p_1	(14, 31)	2	4
p_2	(17, 8)	2	5
p_4	(8, 5)	1	4
p_6	(18, 22)	2	3
p_9	(14, 8)	2	4

Figura 6.25 - Padrão de corte produzido para A17.

Padrão de Corte – A18



Padrão de corte: H2R7M1P5M1R8 V2R3M1R2 V2R3M1R2 H3P9

Aproveitamento: 99.82%

Parâmetro α : 0.25

Produzido na iteração: 614

Placa: (52, 55)

Peças utilizadas: $p_2, p_3, p_5, p_7, p_8, e p_9$

<i>Itens</i>	(c_i, l_i)	x_i	d_i
p_2	(9, 8)	2	4
p_3	(16, 8)	4	4
p_5	(12, 25)	1	4
p_7	(14, 19)	2	4
p_8	(13, 5)	1	4
p_9	(21, 41)	3	4

Figura 6.26 - Padrão de corte produzido para A18.

7. CONCLUSÕES

Neste trabalho, foi realizado um estudo sobre o problema de corte bidimensional guilhotinado e restrito com o intuito de apresentar um método eficiente para a resolução deste problema. Devido à sua natureza combinatória, este é um problema pertencente à classe *NP-Hard* e não se tem conhecimento da existência de algoritmos exatos que, em situações reais, o resolva em tempo polinomial. Na prática uma solução aproximada é suficientemente aceitável. Sendo assim, propõe-se um algoritmo heurístico, baseado na técnica GRASP, capaz de apresentar soluções de boa qualidade em tempo compatível com a necessidade de rapidez presente nos setores produtivos.

Como primeira conclusão, os resultados apresentados com o algoritmo GRASP-2D demonstram que a metodologia GRASP pode ser aplicada com sucesso em problemas de corte bidimensional guilhotinado e restrito.

Para as instâncias da primeira classe, retiradas da literatura, os resultados produzidos com a execução do GRASP-2D foram sempre superiores aos apresentados pelos algoritmos Busca Tabu e *Simulated Annealing*. Em média, a melhoria obtida pelo GRASP-2D, nos padrões de corte gerados para as instâncias desta classe, foi de 1.37% e 3.28% em relação aos algoritmos Busca Tabu e *Simulated Annealing*, nesta ordem.

Quanto às instâncias geradas aleatoriamente, novamente, o GRASP-2D mostrou-se bastante eficiente. Em quatorze instâncias, das dezoito encontradas nesta classe, os respectivos padrões de corte encontrados pelo GRASP-2D foram os melhores possíveis. Os resultados apresentados pelo Busca Tabu também foram muito bons, exceto para as instâncias A14 e A18. Nestas instâncias, a diferença entre os respectivos valores de função objetivo, 3.9% e 5.31%, resultaram em perdas consideráveis. Já o *Simulated Annealing* chegou a apresentar até 8.39% de diferença (instância A18) com relação aos valores obtidos pelo GRASP-2D. As melhorias apresentadas pelo GRASP-2D foram, em média,

1.14% e 3.29%, em relação aos algoritmos Busca Tabu e *Simulated Annealing*, respectivamente.

É importante ressaltar que, em algumas instâncias, o valor atribuído ao parâmetro α foi decisivo, influenciando o resultado final. Como por exemplo, nas instâncias L3, L4 e L5, o GRASP-2D obteve os melhores resultados com α igual a 0.25, 0.25 e 0.1, nesta ordem. Já na segunda classe de instâncias, os melhores padrões de corte foram encontrados pelo GRASP-2D, com α igual a 0.25, 0.1 e 0.25, em A3, A16 e A18, respectivamente. Logo, confirma-se que o valor do parâmetro α tem influência direta na qualidade da solução produzida.

Os tempos computacionais gerados nos testes feitos com o algoritmo proposto foram bons, considerando o número de iterações executadas (1000 iterações). Mas este tempo pode ser melhorado, diminuindo a quantidade máxima de iterações utilizadas como critério de parada na GRASP. Caso não fosse adotada a qualidade de solução como critério de avaliação de desempenho das heurísticas, o tempo de processamento do GRASP-2D poderia ser reduzido consideravelmente. Isto se torna evidente pelo fato de a maioria dos padrões de corte, apresentados pelo GRASP-2D, terem sido gerados com menos de 100 iterações.

Algumas perspectivas para pesquisas futuras seriam:

- Definir uma estratégia de economia de chamadas, com a inclusão de filtros, na fase de melhoria do algoritmo GRASP-2D;
- Desenvolver uma ferramenta heurística híbrida, envolvendo as metaheurísticas GRASP e Busca Tabu, para tratar o problema de corte bidimensional guilhotinado e restrito;
- Criação de um ambiente computacional voltado para ambientes de produção;
- Aplicação da técnica GRASP na otimização de problemas de corte e empacotamento com outras dimensões no Espaço Euclidiano ou dimensões abstratas.

REFERÊNCIAS BIBLIOGRÁFICAS

ALVARENGA, A.G.; DAZA, V.P. Um método de busca informada para o problema do corte de peças. **Anais do IX Congresso Brasileiro de Automática**, Vitória, 1992.

ALVARENGA, A.G.; GOMES, F.N.; PARADA, V. O problema de corte de peças na indústria de calçados. **Anais do VII Congresso Latino-Ibero-Americano de Investigação Operativa**. Santiago, 1994.

AMARAL, A. R. S. **Abordagens via Tabu Search e Simulated Annealing para o problema de corte bi-dimensional restrito**. 1994. Dissertação (Mestrado em Engenharia Elétrica – Automação) – Programa de Pós-Graduação em Engenharia Elétrica, UFES, Vitória.

CAMPELLO, R.; MACULAN, N. **Projeto e análise de algoritmos heurísticos**. Rio de Janeiro: Escola Brasileira de Otimização – EBO, 1989.

CAMPELLO, R.; MACULAN, N. **Algoritmos e heurísticas: desenvolvimento e avaliação de performance**. Niterói: Editora da Universidade Federal Fluminense – EDUFF, 1994.

CARNEIRO, S. A. **Problema de corte restrito, via Algoritmo Genético**. 1994. Dissertação. (Mestrado em Engenharia Elétrica – Automação) – Programa de Pós-Graduação em Engenharia Elétrica, UFES, Vitória.

CHRISTOFIDES, N.; WHITLOCK, C. An algorithm for two-dimensional cutting problems. **Operations Research**, n. 25, p. 30-44, 1977.

CHRISTOFIDES, N.; HADJICONSTANTINOOU, E. An exact algorithm for orthogonal 2-D cutting problems using guillotine cuts. **European Journal of Operational Research**, n. 83, p. 21-38, 1995.

CUNG, V.D.; HIFI, M.; LE CUN, B. Constrained two-dimensional cutting stock problems. A best-first branch-and-bound algorithm. **International Transactions in Operational Research**, n. 7, p. 185-210, 2000.

DAZA, V.P.; ALVARENGA, A.G.; DIEGO, J. Exact solutions for constrained two-dimensional cutting problems. **European Journal of Operational Research**, n. 84, p. 633-644, 1995.

DIAZ, A.; GLOVER, F.; GHAZIRI, H.M.; GONZALEZ, J.L.; LAGUNA, M.; MOSCATO, P.; TSENG, F.T. **Optimización heurística y redes neuronales: en dirección de operaciones e ingeniería**. Madri: Editorial Paraninfo, 1996.

DYCKHOFF, H. A typology of cutting and packing problems. **European Journal of Operational Research**, n. 44, p. 145-149, 1990.

DYCKHOFF, H.; FINKE, U. **Cutting and Packing in Production and Distribution: Typology and Bibliography**. Heidelberg: Springer-Verlag Co., 1992.

FEO, T.A.; RESENDE, M.G.C. A probabilistic heuristic for a computationally difficult set covering problem. **Operation Research Letters**, n. 8, p. 67-71, 1989.

FEO, T.A.; RESENDE, M.G.C. Greedy Randomized Adaptive Search Procedures. **Journal of Global Optimization**, n. 6, p. 109-133, 1995.

GAREY M. R.; JOHSON D. S. **Computers and Intractability: a guide to the teory of NP-Completeness**. New York: W. H. Freeman and Company, 1979.

GILMORE, P.C.; GOMORY, R.E. A linear programming approach to the cutting stock problem. **Operations Research**, n. 9, p. 849-859, 1961.

GILMORE, P.C.; GOMORY, R.E. A linear programming approach to the cutting stock problem - Part II. **Operations Research**, n. 11, p. 863-888, 1963.

GILMORE, P.C.; GOMORY, R.E. Multistage cutting stock problems of two and more dimensions. **Operations Research**, n. 13, p. 94-120, 1965.

GILMORE, P.C.; GOMORY, R.E. The theory and computation of knapsack functions. **Operations Research**, n. 14, p. 1045-1074, 1966.

GILMORE, P.C.; GOMORY, R.E. The theory and computation of knapsack functions. **Operations Research**, n. 15, p. 1045-1075, 1967.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers and Operations Research**. n.13, p. 533-549, 1986.

GLOVER, F.; LAGUNA, M. **Tabu search**. Boston: Kluwer Academic Publishers, 1997.

GOLDBARG, M.C.; LUNA, H.P.L. **Otimização combinatória e programação linear: modelos e algoritmos**. Rio de Janeiro: Editora Campus, 2000.

HERZ, J.C. Recursive computational procedures for two-dimensional stock cutting. **IBM Journal of Research and Development**, n.16, p. 462-469, 1972.

HIFI, M. An improvement of Viswanathan and Bagchi's exact algorithm for cutting stock problems. **Computers & Operations Research**, n.24, p. 727-736, 1997.

KIRKPATRICK, S.; GELLAT, C. D.; VECCHI, M.P. Optimization by Simulated Annealing. **Science**, n. 220, p. 671-680, 1983.

MORABITO, R.; ARENALES, M. N. Staged and constrained two-dimensional guillotine cutting problems: an AND/OR-graph approach. **European Journal of Operational Research**, n. 94, p. 548-560, 1996.

NORONHA, T. F. Uma abordagem sobre estratégias metaheurísticas. **Revista Eletrônica de Iniciação Científica**. Publicação da Sociedade Brasileira de Computação – SBC, São Paulo, ano 1, vol. 1, n. 1, ago. 2001. Disponível em: <<http://www.sbc.org.br/reic/edicoes/2001e1/cientificos/UmaAbordagemSobreEstrategiasMetaheurísticas.pdf>>. Acesso em: 12 out. 2003.

OLIVEIRA, J.F.; FERREIRA, J.S. An improved version of Wang's Algorithm for two-dimensional cutting problems. **European Journal of Operational Research**, n. 44, p. 256-266, 1990.

OLIVEIRA, J.F.; FERREIRA, J.S. A geração rápida de colunas: uma alternativa na técnica de Gilmore e Gomory para problemas de corte. **Investigação Operacional**, Lisboa, v. 13, p. 101-117, 1993.

PARADA, V.; PRADENAS, L.; SOLAR, M. Heurísticas Híbridas em Problemas de Corte de Peças. **Anais do X Congresso Latino-Ibero-Americano de Investigação Operativa**. Cidade do México, 2000.

PINTO, M.; ARENALES, M.N. O problema de corte de estoque bidimensional inteiro. **Anais do XXX Simpósio Brasileiro de Pesquisa Operacional - III Oficina Nacional de Problemas de Corte & Empacotamento**. Curitiba, 1998.

SILVEIRA, R. J. **O problema de corte bidimensional guilhotinado e restrito: aplicação na indústria de circuitos impressos**. 1999. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, UFSCar, São Carlos.

SLACK, N.; CHAMBERS, S.; HARLAND, C.; HARRISON, A.; JOHNSTON, R. **Administração da produção**. 1ª ed. São Paulo: Atlas, 1996.

SOUZA, M. J. Freitas. **Inteligência computacional para otimização**. Apostila da disciplina Inteligência Computacional para Otimização. Universidade Federal de Ouro Preto – UFOP. Ouro Preto, 2002. Disponível em: <<http://www.decom.ufop/prof/marcone>>. Acesso em: 12 out. 2003.

VALDÉS, R.A.; PARAJÓN, A.; TAMARIT, J.M. A tabu search algorithm for large-scale guillotine (un)constrained two-dimensional cutting problems. **Computers & Operations Research**, n.29, p. 925-947, 2002.

VIEIRA NETO, E. **Uma heurística GRASP para o problema do corte unidimensional**. 1999. Dissertação (Mestrado em Ciências das Engenharia) – Centro de Ciências e Tecnologia, Laboratório de Engenharia de Produção – LEPROD, UENF, Campos dos Goytacazes.

VIEIRA NETO, E. **GRASP: efeito da independência das soluções iniciais na otimização do corte unidimensional**. 2004. Tese (Doutorado em Ciências das Engenharia) – Centro de Ciências e Tecnologia, Laboratório de Engenharia de Produção – LEPROD, UENF, Campos dos Goytacazes

WANG, P. Y. Two algorithms for constrained two-dimensional cutting stock problems. **Operations Research**, n. 31, p. 573-586, 1983.

APÊNDICE – Soluções geradas pelo algoritmo GRASP-2D

A seguir são apresentadas algumas das soluções representadas por *strings* de caracteres, obtidas pelo algoritmo GRASP-2D, com os seus respectivos valores de função objetivo (percentual de aproveitamento em relação ao somatório de placas utilizadas), para as instâncias exibidas no Anexo 1.

➤ Instância 1 – L1

Padrão de Corte 1: V2P1M1P3 V1P2M1P5 H2R3 H1P6M1R7

Aproveitamento do Padrão 2: 100.00%

Padrão de Corte 2: H5P4 V1P6M1P5

Aproveitamento do Padrão 2: 50.00%

Aproveitamento da Solução: 75%

➤ Instância 2 – L2

Padrão de Corte 1: H2R1 H4P4 H4P4 H3P2 H3R5

Aproveitamento do Padrão 1: 94.44%

Padrão de Corte 2: H2R1 H2R1 H2R1 H3P2

Aproveitamento do Padrão 2: 100.00%

Padrão de Corte 3: V6P5 V1R4M5R3 V3R2M3R3

Aproveitamento do Padrão 3: 86.27%

Padrão de Corte 4: H1P1M1P3

Aproveitamento do Padrão 4: 17.90%

Aproveitamento da solução: 74.65%

➤ **Instância 3 – L3**

Padrão de Corte 1: H1R2 V1R4M1R7 H1R6M1R10 V1P9 V2P8

Aproveitamento do Padrão 1: 97.32%

Padrão de Corte 2: H1R2 V3P7M1P1 V1P10 V1R3

Aproveitamento do Padrão 2: 82.04%

Padrão de Corte 3: H1R8M1P5 H1R9 H1R5 V1P6

Aproveitamento do Padrão 3: 52.25%

Padrão de Corte 4: H1R2 H1R6 H1R2

Aproveitamento do Padrão 4: 42.32%

Aproveitamento da solução: 68.48%

➤ **Instância 4 – L4**

Padrão de Corte 1: H1P2 V2P9M1P3 V2R5 H1R8 H1P8 H1P8

Aproveitamento do Padrão 1: 96.93%

Padrão de Corte 2: V1R7 V1R6 H2P3 H1R4 V1R8

Aproveitamento do Padrão 2: 70.14%

Padrão de Corte 3: V4P10

Aproveitamento do Padrão 3: 82.28%

Padrão de Corte 4: V1P4M2P1 H2R5

Aproveitamento do Padrão 4: 59.64%

Aproveitamento da solução: 77.25%

➤ **Instância 5 – L5**

Padrão de Corte 1: V3R6M3P10M2P9 H3P3 H3P4 V1P9 H1R7

Aproveitamento do Padrão 1: 97.25%

Padrão de Corte 2: H1R1M1P8 H2R8 V1R5 H1P5 H1P2

Aproveitamento do Padrão 2: 72.89%

Aproveitamento da solução: 85.07%

➤ **Instância 6 – L6**

Padrão de Corte 1: H1P18 V1P13M1P8

Aproveitamento do Padrão 1: 85.93%

Padrão de Corte 2: H1P17 V1R16 V2P3

Aproveitamento do Padrão 2: 91.68%

Padrão de Corte 3: V2R13 H1R7 H1R6 V1P3

Aproveitamento do Padrão 3: 94.43%

Padrão de Corte 4: V1R20M1R10 H1R1

Aproveitamento do Padrão 4: 76.32%

Padrão de Corte 5: V2P12 H1R8 V1P4M1P2

Aproveitamento do Padrão 5: 89.57%

Padrão de Corte 6: V2R14 H1R6 V2P4

Aproveitamento do Padrão 6: 98.36%

Padrão de Corte 7: V2P11M1R10

Aproveitamento do Padrão 7: 73.54%

Padrão de Corte 8: H1P15 V1R19

Aproveitamento do Padrão 8: 77.36%

Padrão de Corte 9: H1P10M1P9 H1R5M1P4M1P2 H2R2

Aproveitamento do Padrão 9: 88.64%

Padrão de Corte 10: H2R7 H2P9 V1P7

Aproveitamento do Padrão 10: 80.14%

Aproveitamento da solução: 85.60%

ANEXO 1 – Instâncias Disponíveis na Literatura

Em seqüência, as tabelas apresentarão as informações referentes às seis instâncias disponíveis na literatura. Estas instâncias possuem as seguintes características gerais:

- Comprimento dos objetos: $15 \leq C \leq 40$.
- Largura dos objetos: $10 \leq L \leq 108$.
- Número de itens diferentes: $7 \leq m \leq 20$.

Instância 1 – L1: Christofides e Whitlock (1977)

$$C = 15, L = 10, m = 7$$

Itens	Demanda	Comprimento	Largura
1	2	8	4
2	1	3	7
3	3	8	2
4	5	3	4
5	2	3	3
6	2	3	2
7	1	2	1

Instância 2 – L2: Wang(1983)

$$C = 60, L = 108, m = 5$$

Itens	Demanda	Comprimento	Largura
1	9	28	30
2	9	20	24
3	9	16	20
4	9	14	21
5	9	12	18

Instância 3 – L3: Oliveira e Ferreira (1990)

$$C = 40, L = 70, m = 10$$

Itens	Demanda	Comprimento	Largura
1	1	29	5
2	4	9	39
3	1	55	9
4	1	31	15
5	2	11	16
6	3	23	21
7	4	29	14
8	3	16	19
9	2	9	36
10	2	22	4

Instância 4 – L4: Oliveira e Ferreira (1990)

$$C = 40, L = 70, m = 10$$

Itens	Demanda	Comprimento	Largura
1	2	22	18
2	1	40	10
3	3	13	27
4	2	23	18
5	4	29	8
6	1	16	4
7	1	47	9
8	4	19	19
9	2	13	16
10	4	36	16

Instância 5 – L5: Christofides e Whitlock (1977)

$$C = 40, L = 70, m = 10$$

Itens	Demanda	Comprimento	Largura
1	1	21	22
2	1	31	13
3	3	9	35
4	3	9	24
5	2	30	7
6	3	11	13
7	1	10	14
8	3	14	8
9	3	12	8
10	3	13	7

Instância 6 – L6: Christofides e Whitlock (1977), Wang (1983)

$$C = 40, L = 70, m = 20$$

Itens	Demanda	Comprimento	Largura
1	1	17	9
2	4	11	19
3	3	12	21
4	4	14	23
5	1	24	15
6	2	24	15
7	4	25	16
8	2	27	17
9	3	18	29
10	3	21	31
11	2	32	22
12	2	23	33
13	3	34	24
14	2	35	25
15	1	36	26
16	1	37	27
17	1	38	28
18	1	39	29
19	1	41	30
20	1	43	31

ANEXO 2 – Instâncias Geradas Aleatoriamente

Nas tabelas seguintes serão apresentados os dados referentes às instâncias geradas aleatoriamente. Estas instâncias possuem as seguintes características gerais:

- Comprimento dos objetos: $10 \leq C \leq 70$.
- Largura dos objetos: $16 \leq L \leq 108$.
- Número de itens diferentes: $7 \leq m \leq 20$.

Instância 1 – A1

$$C = 15, L = 20, m = 5$$

Itens	Demanda	Comprimento	Largura
1	8	3	7
2	8	4	4
3	8	5	3
4	8	6	5
5	8	7	5

Instância 2 – A2

$$C = 11, L = 20, m = 5$$

Itens	Demanda	Comprimento	Largura
1	7	2	10
2	7	4	3
3	7	3	3
4	7	3	9
5	9	6	7

Instância 3 – A3

$$C = 23, L = 56, m = 10$$

Itens	Demanda	Comprimento	Largura
1	5	3	4
2	6	7	4
3	8	7	6
4	6	6	3
5	9	7	7
6	4	9	12
7	6	4	15
8	7	5	4
9	5	3	35
10	5	12	13

Instância 4 – A4

$$C = 10, L = 16, m = 10$$

Itens	Demanda	Comprimento	Largura
1	7	4	3
2	4	4	1
3	4	7	4
4	5	5	1
5	6	7	3
6	7	2	7
7	6	5	2
8	4	4	2
9	5	3	3
10	4	9	1

Instância 5 – A5

$$C = 17, L = 25, m = 10$$

Itens	Demanda	Comprimento	Largura
1	1	3	10
2	3	4	3
3	4	3	9
4	2	8	3
5	2	7	9
6	3	9	5
7	2	6	2
8	4	12	3
9	4	10	5
10	1	9	8

Instância 6 – A6

$$C = 20, L = 23, m = 10$$

Itens	Demanda	Comprimento	Largura
1	4	3	5
2	4	4	6
3	4	2	3
4	4	3	6
5	3	3	11
6	4	5	11
7	3	3	8
8	3	6	8
9	2	13	5
10	2	11	16

Instância 7 – A7

$$C = 20, L = 35, m = 10$$

Itens	Demanda	Comprimento	Largura
1	5	12	4
2	5	9	4
3	5	3	5
4	5	2	8
5	5	2	5
6	5	3	4
7	5	3	12
8	5	17	3
9	5	8	6
10	5	10	6

Instância 8 – A8

$$C = 17, L = 21, m = 10$$

Itens	Demanda	Comprimento	Largura
1	4	3	3
2	4	3	4
3	4	3	7
4	4	5	5
5	4	4	7
6	4	6	4
7	4	2	11
8	4	2	5
9	4	3	2
10	4	5	7

Instância 9 – A9

$$C = 22, L = 35, m = 8$$

Itens	Demanda	Comprimento	Largura
1	4	8	6
2	4	14	4
3	4	6	5
4	4	7	3
5	4	6	6
6	4	6	9
7	4	8	11
8	4	8	15

Instância 10 – A10

$$C = 17, L = 23, m = 9$$

Itens	Demanda	Comprimento	Largura
1	4	7	5
2	4	16	4
3	4	4	6
4	4	13	5
5	4	12	6
6	4	4	2
7	4	10	7
8	4	4	5
9	4	3	7

Instância 11 – A11

$$C = 16, L = 23, m = 9$$

Itens	Demanda	Comprimento	Largura
1	4	3	7
2	4	6	5
3	4	11	3
4	4	9	5
5	4	4	6
6	4	3	4
7	4	6	9
8	4	4	8
9	4	3	6

Instância 12 – A12

$$C = 16, L = 31, m = 10$$

Itens	Demanda	Comprimento	Largura
1	4	7	6
2	4	4	9
3	4	5	6
4	4	3	10
5	4	6	6
6	4	6	4
7	4	10	4
8	4	4	7
9	4	9	6
10	4	4	5

Instância 13 – A13

$$C = 11, L = 21, m = 12$$

Itens	Demanda	Comprimento	Largura
1	4	3	2
2	2	8	1
3	3	9	2
4	3	4	4
5	5	2	4
6	5	1	2
7	2	7	1
8	4	5	3
9	3	7	2
10	4	4	1
11	3	4	3
12	4	6	2

Instância 14 – A14

$$C = 21, L = 22, m = 12$$

Itens	Demanda	Comprimento	Largura
1	4	11	4
2	4	14	3
3	4	5	4
4	4	8	5
5	4	9	2
6	4	5	2
7	4	8	3
8	4	7	4
9	4	3	3
10	4	9	5
11	4	4	2
12	4	4	4

Instância 15 – A15

$$C = 18, L = 42, m = 7$$

Itens	Demanda	Comprimento	Largura
1	4	4	21
2	3	15	4
3	3	6	15
4	4	6	14
5	4	8	9
6	3	10	9
7	3	12	6

Instância 16 – A16

$$C = 30, L = 75, m = 10$$

Itens	Demanda	Comprimento	Largura
1	4	22	24
2	4	8	14
3	4	27	6
4	4	9	10
5	3	8	27
6	4	14	3
7	3	6	17
8	4	13	8
9	4	14	26
10	3	9	26

Instância 17 – A17

$$C = 36, L = 61, m = 9$$

Itens	Demanda	Comprimento	Largura
1	4	14	31
2	5	17	8
3	5	19	22
4	4	8	5
5	3	18	10
6	3	18	22
7	4	26	12
8	4	5	12
9	4	14	8

Instância 18 – A18

$$C = 52, L = 55, m = 10$$

Itens	Demanda	Comprimento	Largura
1	4	16	20
2	3	9	8
3	4	16	8
4	4	12	25
5	4	9	14
6	4	1	5
7	4	14	19
8	4	13	5
9	4	12	41
10	4	10	16

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)