
**Algumas extensões do problema de corte de
estoque com sobras de material aproveitáveis**

Adriana Cristina Cherri

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 17/04/2009

Assinatura: _____

Algumas extensões do problema de corte de estoque com sobras de material aproveitáveis

Adriana Cristina Cherri

Orientador: *Prof Dr Marcos Nereu Arenales*

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional.

USP - São Carlos

Abril/2009

Ao Celso Henrique, meu esposo e aos meus pais, Luiz e Maria, testemunhas de toda minha dedicação por este trabalho...

Agradecimentos

À Deus, por ter sempre me guiado e iluminado.

Ao Professor Marcos Nereu Arenales pela orientação, compreensão, amizade, e principalmente, pela confiança depositada no desenvolvimento deste trabalho.

Aos meus pais Luiz e Maria, pelo amor, compreensão, apoio e dedicação incondicional que sempre tiveram por mim.

Aos meus irmãos César e Henrique, e à toda minha família pelo carinho e incentivo.

Ao meu querido Celso Henrique pelo apoio, compreensão, paciência, amizade, amor e por ter estado sempre ao meu lado.

À professora Andréa Carla Gonçalves Vianna (UNESP - Bauru), por toda sua ajuda e pelos bons momentos vivenciados. Também agradeço aos professores do Departamento de Matemática (UNESP - Bauru) pelo incentivo e amizade.

Ao professor Horacio Hideki Yanasse (INPE), pelas sugestões e dicas que muito enriqueceram nosso trabalho.

À minha família de amigos: Kelly, Carla, Douglas, Claudia, Lana, Pedro, Aline, Victor, Tamara, Guilherme e Márcio pelo apoio, paciência e pelas boas conversas. Também agradeço todos os colegas e professores do Laboratório de Otimização que diferentes maneiras fizeram parte da minha vida durante o doutorado.

Às amigas Juliana e Edmary pela companhia, amizade, carinho e pelos bons momentos de descontração nas aulas de inglês.

Aos professores e funcionários do ICMC-USP que, direta ou indiretamente, contribuíram com este trabalho.

À FAPESP pela credibilidade e apoio financeiro.

Enfim, agradeço a todos que, tanto do lado profissional quanto do lado pessoal, me ajudaram a finalizar mais um capítulo de minha vida. Obrigada a todos!

Resumo

Os problemas de corte de estoque consistem em cortar um conjunto de objetos disponíveis em estoque para produzir um conjunto de itens em quantidades e tamanhos especificados, de modo a otimizar uma função objetivo. Tais problemas têm inúmeras aplicações industriais e têm sido bastante estudados na literatura. Tipicamente, problemas de corte têm como principal objetivo a minimização das sobras. Entretanto, como a qualidade dos padrões de corte depende diretamente dos tamanhos e quantidades dos itens a serem produzidos, nesta tese, consideramos que se a demanda presente gerar sobras indesejáveis (nem tão grandes para serem aproveitáveis, nem tão pequenas para serem perdas aceitáveis), então convém gerar retalhos (não computáveis como perda) que serão utilizados para produzir itens de demandas futuras. Desta forma, algumas características desejáveis para uma boa solução são definidas e alterações em métodos heurísticos clássicos são apresentadas, de modo que os padrões de corte com sobras indesejáveis são alterados. Para os problemas de corte unidimensionais, desenvolvemos procedimentos heurísticos que consideram o aproveitamento de sobras, mantendo como o principal objetivo a minimização das perdas. Outra abordagem para este problema, considera o caso em que além da minimização das perdas, os retalhos disponíveis em estoque devem ter prioridade de uso em relação aos demais objetos durante o processo de corte. A análise do desempenho dos procedimentos heurísticos propostos quando somente a minimização das perdas é considerada, é realizada com base em exemplos da literatura, exemplos práticos e exemplares gerados aleatoriamente. Para os procedimentos heurísticos que priorizam o corte dos retalhos do estoque, além de exemplares da literatura, simulamos uma situação em múltiplos períodos na qual problemas de corte de estoque em sucessivos períodos são resolvidos. A cada período, um problema para o período seguinte é gerado considerando atualizações do estoque, os retalhos gerados nos períodos anteriores e uma nova demanda de itens que é

gerada aleatoriamente. No caso bidimensional, também consideramos problemas em que, além da perda mínima, os retalhos disponíveis em estoque devem ter prioridade de corte em relação aos demais objetos. Para resolver este problema, alterações foram realizadas na abordagem grafo E/OU e em procedimentos heurísticos da literatura. A análise do desempenho dos procedimentos heurísticos propostos considera problemas práticos retirados da carteira de pedidos de uma pequena empresa de esquadrias metálicas. Devido à dificuldade na análise dos procedimentos heurísticos desenvolvidos que consideram o aproveitamento de sobras (as soluções apresentam características importantes e conflitantes), também apresentamos neste trabalho uma estratégia *fuzzy* para facilitar a análise das soluções obtidas. Os testes computacionais são realizados considerando os procedimentos heurísticos desenvolvidos para os problemas de corte unidimensionais com sobras aproveitáveis e problemas gerados aleatoriamente.

Palavras-chave: problema de corte de estoque unidimensional e bidimensional, aproveitamento das sobras de material.

Abstract

Cutting stock problems consist of cutting a set of available objects in order to produce ordered items in specified amounts and sizes, in such way to optimize an objective function. Such problems have a great number of industrial applications and are widely studied in the literature. Typically, cutting problems have as main objective the minimization of the leftovers. However, since the cutting patterns quality depends directly of the sizes and amounts of the items that will be produced, in this tesis, we consider that if the present demand to generate undesirable waste (not large enough to be used, nor too small to be acceptable waste), then it is better to generate retails (not computed as waste) that will be used to produce items to meet future demands. In this way, some desirable characteristics for a good solution are defined and alterations in classical heuristic methods are presented, such that the cutting patterns with undesirable waste are altered. To the one-dimensional cutting stock problems, we developed heuristic procedures that consider the usable leftovers and preserve as main objective the minimization of the waste. Other approach for this problem considers the case in witch, beside minimal waste, the available retails in stock must be used with priority in relation to the other objects during the cutting process. The performance of the modified heuristics procedures, when only the minimal waste is considered, is observed by solving instances from the literature, practical instances and randomly generated instances. For heuristic procedures that prioritize the cut of retails of the stock, beside the instances from the literature, we simulated a situation in multiple periods in that cutting stock problems in successive periods are solved. In each period, a problem to the next period is generated considering updating of the stock, the retails generated in previous periods and a new demand of items that is randomly generated. For the two-dimensional cutting problems, we also consider problems in that, beside minimization of the waste, the available retails in stock must be used with priority

in relation to the other objects. To solve this problem, alterations were realized in an AND/OR graph approach and in heuristic procedures of the literature. The performance of the proposed heuristics procedures is observed by solving practical instances provided by a small metallic frameworks industry. Due to difficulty in analyze the heuristic procedures developed for the cutting stock problem with usable leftover (the solutions present important and conflicting characteristics), we also present a *fuzzy* strategy to facilitate the analysis of the obtained solutions. The computational results are realized considering the developed heuristic procedures to the one-dimensional cutting stock problem with usable leftover and randomly generated instances.

Key-words: one-dimensional and two-dimensional cutting stock problems, usable leftovers.

Índice

| | |
|--|-------------|
| Lista de Figuras | xiii |
| Lista de Tabelas | xv |
| 1 Introdução | 1 |
| 1.1 Classificação dos problemas de corte | 5 |
| 1.1.1 Problema de corte unidimensional | 6 |
| 1.1.2 Problemas de corte bidimensional | 6 |
| 1.1.3 Problema de corte tridimensional | 7 |
| 1.2 Objetivos | 7 |
| 1.3 Organização do Texto | 7 |
| 2 Revisão Bibliográfica | 9 |
| 2.1 Revisão dos problemas de corte unidimensionais com sobras aproveitáveis . | 12 |
| 2.1.1 Notação Global | 13 |
| 2.1.2 Abordagens orientadas a itens | 14 |
| 2.1.3 Abordagens orientadas a padrões | 28 |
| 2.1.4 Metaheurística | 42 |
| 3 O problema de corte de estoque com sobras de material aproveitáveis | 47 |
| 3.1 Problema de corte unidimensional com sobras aproveitáveis - primeira abordagem | 47 |
| 3.2 Problema de corte unidimensional com sobras aproveitáveis e prioridade no uso dos retalhos - segunda abordagem | 52 |
| 3.3 Problema de corte bidimensional com sobras aproveitáveis | 54 |

| | | |
|----------|--|------------|
| 4 | Métodos de solução para o problema de corte unidimensional com sobras aproveitáveis | 59 |
| 4.1 | Heurísticas Construtivas | 59 |
| 4.1.1 | Procedimento FFD | 60 |
| 4.1.2 | Procedimento Guloso | 60 |
| 4.1.3 | Heurística FFD_A | 61 |
| 4.1.4 | Heurística $Gulosa_A$ | 64 |
| 4.1.5 | Heurística FFD_A^P | 67 |
| 4.2 | Heurísticas Residuais | 67 |
| 4.2.1 | Heurísticas Residuais FFD, Gulosa, FFD_A e $Gulosa_A$ | 71 |
| 4.2.2 | Heurísticas Residuais de Arredondamento Guloso (RAG) | 71 |
| 4.2.3 | Heurísticas RAG_A | 72 |
| 4.2.4 | Heurística RAG_A^P | 74 |
| 4.3 | Exemplo | 75 |
| 5 | Métodos de solução para o problema de corte bidimensional com sobras aproveitáveis | 79 |
| 5.1 | Abordagem grafo E/OU | 79 |
| 5.1.1 | Corte Guilhotinado | 80 |
| 5.1.2 | Grafo E/OU | 82 |
| 5.1.3 | Limitantes inferior e superior | 85 |
| 5.1.4 | Algoritmo Branch & Bound | 86 |
| 5.1.5 | Heurísticas | 87 |
| 5.2 | Abordagem grafo E/OU para o aproveitamento de sobras | 88 |
| 5.3 | Heurística $Gulosa_A^{2D}$ | 96 |
| 5.4 | Heurística RAG_A^{2D} | 98 |
| 5.5 | Exemplo | 100 |
| 6 | Experimentos computacionais | 109 |
| 6.1 | Problema de corte unidimensional com sobras aproveitáveis | 109 |
| 6.1.1 | Resultados de exemplares da literatura | 110 |
| 6.1.2 | Resultados de exemplares gerados aleatoriamente | 123 |

| | | |
|----------|---|------------|
| 6.2 | Problema de corte com sobras aproveitáveis e prioridade no uso de retalhos | 134 |
| 6.2.1 | Resultado de exemplares gerados aleatoriamente | 140 |
| 6.2.2 | Resultados de exemplares da literatura | 144 |
| 6.3 | Problema de corte bidimensional com sobras aproveitáveis | 146 |
| 7 | Abordagem <i>Fuzzy</i> para o problema de corte com sobras aproveitáveis | 151 |
| 7.1 | Definição dos parâmetros | 151 |
| 7.2 | Técnicas de inferência <i>fuzzy</i> | 154 |
| 7.2.1 | Relações de implicação | 156 |
| 7.2.2 | Métodos de defuzzificação | 158 |
| 7.3 | Experimentos computacionais | 158 |
| 7.4 | Análise das soluções | 161 |
| 8 | Conclusões e Propostas Futuras | 167 |
| | Referências Bibliográficas | 169 |
| | Apêndice A. Soluções das classes de exemplares gerados aleatoriamente | 179 |

Lista de Figuras

| | | |
|------|---|-----|
| 1.1 | (a) Objeto (barra) a ser cortado; (b) Padrão de corte produzindo 4 itens e uma perda. | 6 |
| 1.2 | (a) Placa a ser cortada; (b) Padrão de corte produzindo 8 itens e uma perda. | 6 |
| 1.3 | (a) Contêiner; (b) 6 caixas empacotadas no contêiner. | 7 |
| 3.1 | Dados de um problema de corte unidimensional e soluções alternativas. | 49 |
| 3.2 | Classificação das sobras. | 51 |
| 3.3 | Dados de um problema de corte unidimensional e soluções alternativas. | 54 |
| 3.4 | Dados de um problema de corte bidimensional e soluções alternativas. | 58 |
| 5.1 | (a) Corte guilhotinado horizontal, (b) Padrão de corte guilhotinado. | 80 |
| 5.2 | Padrão de corte guilhotinado 2-estágios. | 80 |
| 5.3 | Combinação linear dos comprimentos ($x = \ell_1 + 2\ell_2$). | 81 |
| 5.4 | (a) Grafo E/OU com um caminho particular em destaque; (b) Padrão de corte correspondente. | 83 |
| 5.5 | Grafo E/OU representando padrões de corte alternativos. | 84 |
| 5.6 | Solução gerada pelo Grafo E/OU e análise dos nós finais. | 88 |
| 5.7 | Localização das sobras em um nó final. | 90 |
| 5.8 | (a) padrão de corte gerado pelo grafo E/OU; (b) padrão de corte com alterações. | 91 |
| 5.9 | Gulosa $_A^{2D}$: primeiro padrão de corte. | 102 |
| 5.10 | Gulosa $_A^{2D}$: segundo padrão de corte. | 102 |
| 5.11 | Gulosa $_A^{2D}$: terceiro padrão de corte. | 103 |
| 5.12 | Gulosa $_A^{2D}$: quarto padrão de corte. | 103 |
| 5.13 | Gulosa $_A^{2D}$: quinto padrão de corte. | 104 |
| 5.14 | RAG $_A^{2D}$: primeiro padrão de corte. | 104 |

| | | |
|------|--|-----|
| 5.15 | RAG_A^{2D} : segundo padrão de corte. | 105 |
| 5.16 | RAG_A^{2D} : terceiro padrão de corte. | 105 |
| 5.17 | RAG_A^{2D} : quarto padrão de corte. | 106 |
| 5.18 | RAG_A^{2D} : quinto padrão de corte. | 106 |
| 5.19 | RAG_A^{2D} : sexto padrão de corte. | 107 |
| 5.20 | RAG_A^{2D} : sétimo padrão de corte. | 107 |
| 6.1 | Comprimento da perda e número de retalhos por período. | 139 |
| 6.2 | Perda e número médio de retalhos por período - Itens pequenos. | 141 |
| 6.3 | Perda e número médio de retalhos por período - itens médios. | 142 |
| 6.4 | Perda e número médio de retalhos por período - itens variados. | 143 |
| 6.5 | Perda e número total de retalhos por período. | 148 |
| 7.1 | Sistema <i>Fuzzy</i> | 156 |
| 7.2 | Padrões geométricos: (a) Perda Relativa nos Objetos Padronizados, (b) Retalho Líquido Relativo, (c) Perda Relativa nos Objetos não Padronizados. | 162 |
| 7.3 | Saída <i>Fuzzy</i> - Índice de Aceitabilidade. | 162 |
| 7.4 | Ilustração gráfica do processo de Inferência <i>Fuzzy</i> | 164 |
| 7.5 | Região ativada: Regra 1. | 165 |
| 7.6 | Região ativada: Regra 2. | 165 |
| 7.7 | Ilustração gráfica do processo de Inferência <i>Fuzzy</i> : Saída. | 165 |

Lista de Tabelas

| | | |
|------|--|-----|
| 4.1 | Dados dos objetos em estoque | 76 |
| 4.2 | Dados dos itens a serem produzidos | 76 |
| 4.3 | Solução das heurísticas construtivas | 77 |
| 4.4 | Solução das heurísticas residuais | 77 |
| 5.1 | Dados dos objetos em estoque | 101 |
| 5.2 | Dados dos itens a serem produzidos | 101 |
| 6.1 | Dados do Exemplo 1: Itens | 111 |
| 6.2 | Solução do Exemplo 1 | 112 |
| 6.3 | Dados do exemplo 2: Itens | 112 |
| 6.4 | Solução do Exemplo 2 | 113 |
| 6.5 | Dados do exemplo 3: Itens | 113 |
| 6.6 | Solução do Exemplo 3 | 114 |
| 6.7 | Dados do exemplo 4: Itens | 114 |
| 6.8 | Solução do Exemplo 4 | 115 |
| 6.9 | Dados do exemplo 5: Itens | 115 |
| 6.10 | Solução do Exemplo 5 | 116 |
| 6.11 | Dados do exemplo 6: Itens | 116 |
| 6.12 | Solução do Exemplo 6 | 117 |
| 6.13 | Dados do exemplo 7: Itens | 117 |
| 6.14 | Solução do Exemplo 7 | 118 |
| 6.15 | Dados do exemplo 8: Itens | 118 |
| 6.16 | Solução do Exemplo 8 | 119 |
| 6.17 | Dados do exemplo 9: Itens | 120 |

| | | |
|------|--|-----|
| 6.18 | Solução do Exemplo 9 | 120 |
| 6.19 | Dados do exemplo 10: Itens | 121 |
| 6.20 | Solução do Exemplo 10 | 121 |
| 6.21 | Dados do exemplo 11: Itens | 122 |
| 6.22 | Solução do Exemplo 11 | 122 |
| 6.23 | Descrição das classes | 125 |
| 6.24 | Número médio e comprimento médio total de objetos não padronizados usados . | 127 |
| 6.25 | Número médio e comprimento médio total de objetos padronizados usados . . . | 127 |
| 6.26 | Perda total média, comprimento médio da perda nos objetos padronizados e não padronizados | 129 |
| 6.27 | Comprimento médio total dos retalhos, comprimento médio dos retalhos nos objetos padronizados e não padronizados | 129 |
| 6.28 | Número médio de objetos cortados com: retalho, perda pequena e perda não tão pequena | 131 |
| 6.29 | Porcentagem média de objetos cortados com: retalho, perda pequena e perda não tão pequena | 131 |
| 6.30 | Tempo médio e execução (em segundos) | 133 |
| 6.31 | Classificação das heurísticas | 133 |
| 6.32 | Descrição das classes | 136 |
| 6.33 | Dados da demanda | 136 |
| 6.34 | Solução para o 1º período | 137 |
| 6.35 | Dados do estoque - Heurística RAG | 137 |
| 6.36 | Dados do estoque - Heurística RAG_A | 138 |
| 6.37 | Dados do estoque - Heurística RAG_A^P | 138 |
| 6.38 | Solução para o 2º período | 138 |
| 6.39 | Solução depois de 12 períodos | 139 |
| 6.40 | Solução média - Itens pequenos | 141 |
| 6.41 | Solução média - Itens médios | 142 |
| 6.42 | Solução média - Itens variados | 143 |
| 6.43 | Resultado computacional para os 320 exemplares | 145 |
| 6.44 | Solução para 12 períodos | 147 |

| | | |
|-----|--|-----|
| 7.1 | Descrição das classes | 159 |
| 7.2 | Perda média relativa nos objetos padronizados, não padronizados e retalho líquido relativo | 160 |
| 7.3 | Regras Fuzzy | 163 |
| 7.4 | Classificação das Soluções | 166 |

Capítulo 1

Introdução

O problema de corte de estoque consiste em cortar um conjunto de peças disponíveis em estoque para a produção de um conjunto de itens em quantidades e tamanhos especificados, otimizando uma determinada função objetivo. Exemplos de funções objetivos incluem, minimizar a perda, minimizar o custo de cortar objetos, minimizar o número total de objetos a serem cortados, maximizar o lucro, minimizar os custos de produção, entre outros.

Problemas de corte podem aparecer em diversos processos industriais, em que os objetos correspondem a barras de aço, bobinas de papel e alumínio, placas metálicas e de madeira, placas de circuito impresso, chapas de vidro e fibra de vidro, peças de couro, etc. Nessas indústrias, a redução dos custos de produção e a melhoria da eficiência estão, frequentemente, associadas à utilização de estratégias adequadas de cortes.

A importância econômica e operacional de tais problemas, bem como a dificuldade de resolução e os trabalhos de Kantorovich (1960) e Gilmore e Gomory, (1961, 1963), despertaram grande interesse na comunidade de pesquisadores da pesquisa operacional por problemas de corte. Centenas de artigos podem ser encontrados na literatura, conforme indicam os artigos de revisão e edições especiais: Hinxman (1980), Dyckhoff *et al.* (1985), Dyckhoff (1990), Dyckhoff e Wäscher (1990), Dyckhoff e Finke (1992), Dowsland e Dowsland (1992), Sweeney e Paternoster (1992), Martello (1994a), Martello (1994b), Bischoff e Wäscher (1995), Dyckhoff *et al.* (1997), Arenales *et al.* (1999), Wang e Wäscher (2002), Hifi (2002), Oliveira e Wäscher (2007) e Wäscher *et al.* (2007). Referências adicionais

podem ser encontradas em ESICUP (2008).

São várias as situações em que surgem os problemas de corte de estoque, cada um deles com suas especificidades, restrições e objetivos definidos pelas exigências práticas impostas em cada ambiente em que estes problemas aparecem.

Um problema que vem sendo recentemente estudado consiste em aproveitar sobras de padrões de corte desde que estas não sejam pequenas. O problema de aproveitamento de sobras foi citado por Brown (1971), entretanto, só mais tarde começaram a aparecer os primeiros trabalhos que abordam este tema.

Dyckhoff (1981) propôs um modelo matemático para o problema de corte, o qual não utiliza geração de colunas para a formação dos padrões de corte e aponta a possibilidade deste modelo ser adaptado e utilizado para tratar o aproveitamento de sobras. A vantagem do modelo proposto é a possibilidade de controlar sobras que surgem durante o processo de corte atribuindo custos a elas. O problema de corte de estoque com sobras aproveitáveis também foi estudado por Arbibi *et al.* (2002) em uma indústria automobilística. Nesta indústria, os itens cortados eram utilizados para a fabricação de correias e apresentavam a mesma altura e comprimentos diferentes. Desta forma, os retalhos gerados durante o processo de corte poderiam ser costurados e utilizados na confecção de determinados itens. Com esta possibilidade de reuso, uma considerável quantidade de material poderia ser economizada. Entretanto, este caso não é considerado nesta tese, ou seja, um retalho somente pode ser cortado para a produção de itens.

Embora o estudo sobre problemas de corte com aproveitamento de sobras seja recente, alguns trabalhos foram publicados. Para considerar o aproveitamento de sobras, Scheithauer (1991) modificou o problema proposto por Gilmore e Gomory (1963). Roodman (1986) com o objetivo de minimizar as perdas considerou um problema com vários tipos de objetos em estoque. Após o processo de corte, as sobras eram estocadas para serem utilizadas posteriormente. Similar a este problema, Sinuany-Stern e Weiner (1994) estudaram o problema de corte unidimensional, com dois objetivos: minimizar a sobra (objetivo considerado mais importante) e acumular a máxima quantidade de sobras no último objeto a ser cortado. A sobra acumulada, desde que fosse maior que o comprimento do menor item demandado, seria utilizada para atender futuras demandas.

Com o objetivo de criar um plano de corte unidimensional para diminuir a perda ou então concentrá-las em um único objeto, Gradisar *et al.* (1997) apresentaram um procedimento heurístico (denominado COLA) para otimizar o corte de rolos em uma indústria de tecidos. Gradisar *et al.* (1999a) propuseram uma modificação no algoritmo COLA (denominado CUT) e em 1999b Gradisar *et al.* desenvolveram uma abordagem que combina programação linear e procedimento heurístico para resolver o problema de corte com sobras aproveitáveis. O propósito desta combinação é a habilidade para atender a demanda dos itens na quantidade solicitada e acumular sobras para que estas possam ser utilizadas no futuro. Chu e Antonio (2007) abordaram o problema de corte com sobras aproveitáveis em uma indústria especializada no corte de metais. Para resolver este problema, consideraram um problema típico de corte unidimensional com algumas restrições técnicas que surgem na indústria de corte de metal.

Gradisar e Trkman (2005) desenvolveram um algoritmo para encontrar a solução para problemas gerais de corte de estoque quando todos os objetos possuem comprimentos diferentes, partindo da solução obtida pelo algoritmo CUT e refazendo padrões que não satisfaziam alguns critérios. Trkman e Gradisar (2007) destacaram a importância de adaptar adequadamente métodos que consideram o aproveitamento de sobras para que não haja acúmulo de retalhos no estoque e propuseram um método de solução que considera esta restrição.

Kos e Duhovnik (2002) também incluíram em seus estudos a possibilidade de utilizar retalhos no atendimento de demandas subsequentes desde que estes tivessem comprimentos superiores ao estimado pelo tomador de decisões com base no tipo e planejamento da produção. Para resolver este problema, os autores utilizaram um algoritmo genético híbrido que minimiza a sobra de material, entretanto, se a sobra possui comprimento suficientemente grande, retorna ao estoque para uso no futuro.

Abuabara e Morabito (2008) e (2009) modificaram o modelo matemático proposto por Gradisar *et al.* (1997) reduzindo o número de restrições e variáveis do modelo. O problema de corte de estoque com sobras aproveitáveis também foi estudado por Cherri *et al.* (2009) em que heurísticas bem conhecidas da literatura (construtivas e residuais) foram modificadas, de modo que as sobras geradas em cada padrão de corte deveriam

ser pequenas para serem descartadas como perdas ou suficientemente grandes para serem estocadas como retalhos os quais seriam utilizados no atendimento de futuras demandas. Koch *et al.* (2008) também consideram o aproveitamento de sobras em um estudo desenvolvido na indústria de madeira. Uma ferramenta de suporte de decisão foi desenvolvida baseada em um modelo de programação linear inteiro.

Detalhes de todos os artigos que consideram o aproveitamento de sobras são apresentados na Seção 2.1. Para cada trabalho, apresentamos a aplicação do estudo desenvolvido, modelagem matemática (quando existir) e alguns comentários sobre os resultados computacionais obtidos.

Além de trabalhos publicados na literatura, o aproveitamento de sobras para problemas de corte unidimensionais também é considerado em alguns softwares comerciais, muito deles com versões livres limitadas disponíveis na internet (1D Nesting Optimizer, Ace Cutting Optimiser, 1D Stock Cutter, Bar Cut Optimizer & Manager, Corte Certo 1D, Cut Logic 1D, Cutting Line, Cutting Optimize, Pipe Cutting Suíte, Plus 1D, Real Cut 1D). Usualmente a minimização da perda é o principal objetivo a ser considerado pelos softwares, entretanto várias restrições adicionais podem ser inseridas.

No caso bidimensional, desconhecemos trabalhos da literatura que abordam o aproveitamento de sobras, entretanto, uma relação de softwares que consideram este problema pode ser encontrada em Macedo *et al.* (2008).

Neste trabalho, além de apresentarmos detalhes do problema de corte com sobras aproveitáveis, consideramos o caso em que a minimização das perdas não é o único objetivo perseguido, ou seja, os retalhos também devem ter prioridades e não podem permanecer muito tempo em estoque. Para priorizar o uso dos retalhos em relação aos objetos padronizados (objetos comprados pela empresa), procedimentos heurísticos desenvolvidos anteriormente por Cherri *et al.* (2009) foram alterados para levar em consideração esta prioridade. Para análise dos novos procedimentos heurísticos desenvolvidos simulamos uma situação em múltiplos períodos na qual problemas de corte de estoque em sucessivos períodos são resolvidos. A cada período, um problema para o período seguinte é gerado considerando atualizações do estoque, retalhos gerados nos períodos anteriores (os quais têm o uso estimulado a cada período) e uma nova demanda de itens que é gerada

aleatoriamente.

Com a finalidade de simplificar e tornar mais precisa a análise das soluções obtidas pelos procedimentos heurísticos desenvolvidos para resolver o problema de corte de estoque com sobras aproveitáveis, algumas técnicas da Lógica *Fuzzy* foram estudadas e utilizadas para realizarmos a comparação e a classificação das soluções heurísticas propostas por Cherri *et al.* (2009).

Outro estudo desenvolvido neste trabalho é o aproveitamento de sobras para os problemas de corte bidimensionais. A literatura para problemas de corte de estoque bidimensionais é vasta (Herz (1972), Christofides e Whitlock (1977), Wang (1983), Beasley (1985), Riehme (1996), Morabito (1992), Arenales e Morabito (1995), Christofides e Hadjiconstantinou (1995), Morabito e Arenales (1994),(1996), Hifi (1997), Vianna (2000), Lodi *et al.* (2002), Hifi (2004), entre outros), entretanto, não encontramos trabalhos que tratam o aproveitamento de sobras de material para o caso bidimensional, apesar de ser uma prática comum em indústrias de pequeno porte.

1.1 Classificação dos problemas de corte

Para classificar os vários tipos de problemas de corte e empacotamento existentes na literatura, Dyckhoff (1990) desenvolveu uma tipologia abrangente, integrando estes problemas. A tipologia foi fundada com base na estrutura lógica dos vários tipos de problemas de corte e empacotamento com o objetivo de unificar o uso de diferentes notações na literatura e concentrar mais adiante pesquisas em tipos especiais de problemas. Entretanto, uma mesma classe poderia abrigar problemas de grande diversidade, sendo que modelos e métodos adequados a um problema não se estendem a outro da mesma classe. Desta forma, Wäscher *et al.* (2007) apresentaram modificações na tipologia de Dyckhoff, além disso, introduziram uma nova categoria que define problemas diferentes dos apresentados anteriormente. A nova tipologia é baseada nas idéias originais de Dyckhoff (1990), porém permite, segundo os autores, que cada problema tenha uma única representação.

A seguir, de forma simplificada, usamos apenas as dimensões dos problemas para classificá-los.

1.1.1 Problema de corte unidimensional

O problema de corte unidimensional envolve apenas uma das dimensões relevante no processo de corte. Problemas de corte unidimensional ocorrem no processo de corte de barras de aço com a mesma seção transversal, bobinas de papel, tubos para produção de treliças, etc. A Figura 1.1 ilustra este tipo de problema:

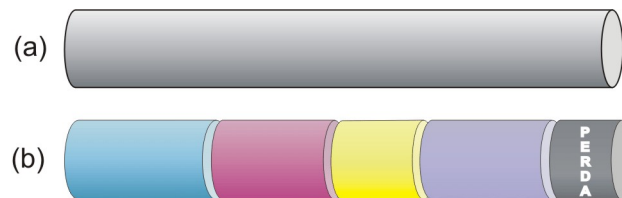


Figura 1.1: (a) Objeto (barra) a ser cortado; (b) Padrão de corte produzindo 4 itens e uma perda.

1.1.2 Problemas de corte bidimensional

No problema de corte bidimensional duas dimensões (comprimento e largura) são relevantes no processo de corte, uma vez que todas as peças cortadas têm a mesma espessura. Resolver este tipo de problema, consiste em combinar geometricamente os itens ao longo do comprimento e da largura dos objetos em estoque. Problemas de corte bidimensional podem ocorrer em indústrias de placas de vidro, alumínio, madeira, etc. A Figura 1.2 ilustra este tipo de problema.

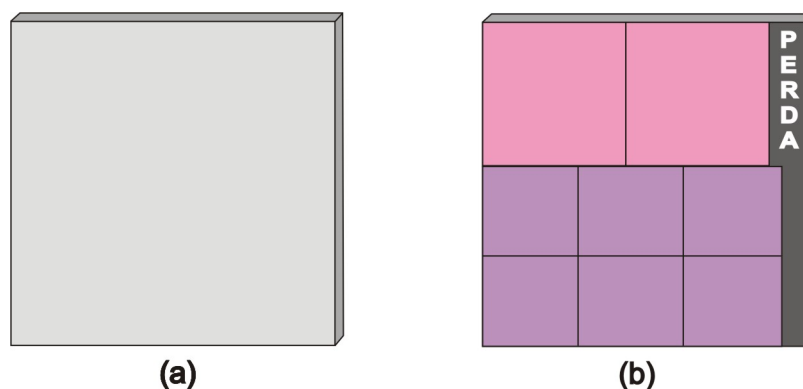


Figura 1.2: (a) Placa a ser cortada; (b) Padrão de corte produzindo 8 itens e uma perda.

1.1.3 Problema de corte tridimensional

No problema de corte tridimensional três dimensões são relevantes no processo de corte. Uma observação importante é certa analogia existente entre o problema de corte tridimensional e o problema de empacotamento: cortar itens de um objeto pode ser visto como empacotar estes itens dentro de um objeto. No caso tridimensional, isto aplica-se ao transporte de carga ferroviária, empacotamento de contêineres, etc. A Figura 1.3 ilustra este tipo de problema.

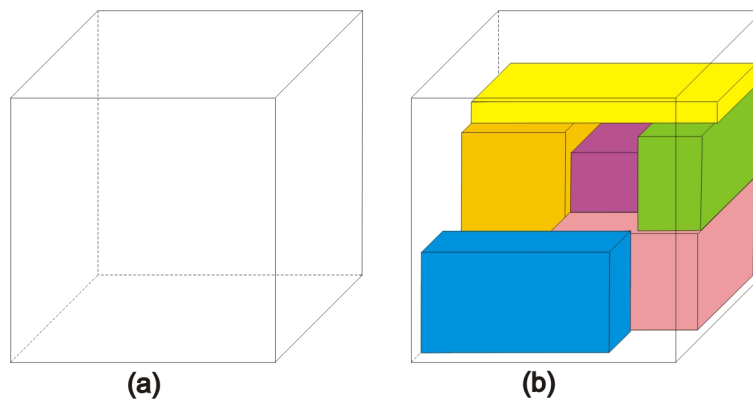


Figura 1.3: (a) Contêiner; (b) 6 caixas empacotadas no contêiner.

1.2 Objetivos

Os objetivos desta tese consistem em propor e analisar métodos de solução para o problema de corte com sobras aproveitáveis, unidimensional e bidimensional.

Para isto, serão utilizadas heurísticas construtivas, geração de colunas, busca em grafo E/OU, lógica *fuzzy* entre outra técnicas.

1.3 Organização do Texto

O texto está dividido em 8 capítulos e 1 apêndice de acordo com a seguinte estrutura:

No Capítulo 2 uma revisão bibliográfica dos problemas de corte, orientada para o problema de aproveitamento é realizada.

No Capítulo 3, definimos o problema de corte de estoque com sobras aproveitáveis no caso unidimensional (quando o objetivo principal é a minimização das perdas e quando além da minimização das perdas há prioridades em utilizar retalhos que ficam disponíveis em estoque) e bidimensional.

Nos Capítulos 4 e 5, descrevemos os métodos de solução utilizados para resolver o problema de corte com sobras aproveitáveis. Apresentamos as alterações realizadas nos procedimentos heurísticos construtivos e residuais, assim como as alterações realizadas no grafo E/OU.

Experimentos computacionais obtidos pelos procedimentos heurísticos desenvolvidos para o caso unidimensional e bidimensional são apresentados no Capítulo 6.

Devido à dificuldade em analisarmos o desempenho das heurísticas propostas para resolver o problema de corte com aproveitamento (muitos fatores devem ser analisados simultaneamente), no Capítulo 7, apresentamos uma abordagem *fuzzy* para realizar uma classificação das heurísticas no caso unidimensional.

O Capítulo 8 é dedicado às conclusões e perspectivas dos próximos passos para continuidade deste trabalho.

No Apêndice A, são apresentadas as tabelas com as soluções médias obtidas para as classes de exemplos geradas aleatoriamente no caso unidimensional.

Capítulo 2

Revisão Bibliográfica

Os problemas de corte de estoque começaram a ser estudados por volta de 1940, entretanto, apenas alguns anos mais tarde surgiram as principais pesquisas nesta área.

Na década de 60 vários trabalhos importantes foram publicados. As modelagens e métodos de resolução de maior repercussão na literatura foram os publicados por Gilmore e Gomory (1961), (1963) e (1965). Em 1961, Gilmore e Gomory apresentaram o método simplex com geração de colunas para um modelo de otimização linear (uma aproximação para o problema) que, pela primeira vez, resolveu problemas reais de corte unidimensional. Em Gilmore e Gomory (1963), um novo método para o problema da mochila foi apresentado e um estudo de caso no corte de papel foi realizado.

Haessler (1975) apresentou um procedimento heurístico computacionalmente eficiente, para resolver problemas de corte de estoque unidimensional com custo fixo associado à troca de padrões de corte. Em Haessler (1980), mudanças foram propostas nos procedimentos de Gilmore e Gomory (1961) e (1963) que geram a solução inicial e os padrões subsequentes a entrar na base. Segundo o autor, controlar a geração dos padrões com limitação do número de itens ajuda a reduzir problemas de arredondamento e mudanças de padrões de corte.

Hinxman (1980) faz uma revisão dos problemas e métodos de resolução dos problemas de corte, quando formaliza a heurística de repetição exaustiva (neste trabalho chamamos de heurísticas construtivas), bastante utilizada na prática, principalmente quando a demanda dos itens é baixa.

Iniciando os anos 90, Stadtler (1990), realizou um estudo de caso em uma indústria de alumínio, com o objetivo de minimizar o número de peças em estoque necessárias para atender a demanda dos clientes. Neste trabalho, o autor observou que os resultados obtidos pela heurística FFD (*First Fit Decreasing*) não eram satisfatórios, então apresentou um novo método, baseado no processo de geração de colunas, proposto por Gilmore e Gomory (1961) e (1963), acrescido de um procedimento de arredondamento para obtenção de uma solução inteira (heurística residual).

Vahrenkamp (1996) fez um estudo comparativo entre o algoritmo de Gilmore e Gomory para a solução do problema de corte de estoque unidimensional com uma heurística baseada em algoritmos genéticos. A estrutura randômica dos algoritmos genéticos leva a uma escolha randômica dos padrões de corte, com o objetivo de minimizar a perda de material.

Wäscher e Gau (1996) reuniram em um artigo vários métodos heurísticos para a resolução do problema de corte de estoque inteiro unidimensional e realizaram um estudo computacional. Recentemente, Poldi e Arenales (2009) também estudaram o problema de obtenção de uma solução inteira para o problema de corte de estoque unidimensional com ênfase para baixas demandas e vários tamanhos de objetos em estoque. Neste trabalho, os autores apresentaram procedimentos de arredondamento gulosos que se mostraram superiores aos procedimentos de arredondamento simples e heurísticas construtivas.

Considerando o problema de corte de estoque bidimensional, em Gilmore e Gomory (1965), um estudo para problemas de corte em duas dimensões ou mais foi desenvolvido. Neste trabalho, foram impostas algumas restrições como corte guilhotinado, estagiado e irrestrito. Tais restrições, entre outras, são encontradas na prática com frequência.

Em 1972, Herz propôs um eficiente algoritmo recursivo para melhorar o tempo computacional das soluções para os problemas de corte de estoque bidimensional, que seria a base da abordagem grafo E/OU proposta por Morabito (1989). Christofides e Whitlock (1977) apresentaram uma árvore de busca para problemas de corte de estoque bidimensionais, na qual existiam restrições na quantidade de peças que seriam produzidas (problema restrito). O método de solução proposto incorpora programação dinâmica e uma rotina de transporte em um algoritmo de busca.

Wang (1983) apresentou dois métodos que constroem padrões de corte guilhotinado restritos, os quais combinam itens demandados ou sub-retângulos, dois a dois por composição de novos sub-retângulos. Cada algoritmo desenvolvido utiliza um parâmetro para limitar as porcentagens máximas de perdas que são geradas em cada sub-retângulo. Mais tarde, Oliveira e Ferreira (1990) e Parada *et al.* (1995) estenderam o algoritmo de Wang. Em 1985, Beasley modelou o problema de corte bidimensional não guilhotinado e restrito como um problema 0-1. Viswanathan e Bagchi (1993) apresentaram o método de busca ‘*best-first*’ baseado em Wang (1983).

Uma abordagem para resolver problemas de corte bidimensional irrestrito guilhotinado foi proposta por Morabito (1989) e Morabito *et al.* (1992). A estratégia consiste em representar o espaço de soluções como caminhos completos em um grafo E/OU (em que os nós representam retângulos e os arcos-E representam cortes guilhotinados). Alguns procedimentos de busca em grafo E/OU foram propostos para a obtenção de soluções e heurísticas foram desenvolvidas para viabilizar computacionalmente o método. Outros trabalhos foram desenvolvidos para a resolução de outros problemas de corte bidimensional (restrito, não guilhotinado, estagiado) e tridimensional usando a abordagem em grafo E/OU para a representação dos padrões de corte (Arenales e Morabito (1995), Morabito e Arenales (1994) e Morabito e Arenales (1996)).

Vianna (2000) estendeu esta abordagem em grafo E/OU para diferentes processos de corte, utilizando limitantes inferiores diferentes aos propostos por Morabito e Arenales (1996). Recentemente, Poldi *et al.* (2005), realizaram um estudo de caso em uma indústria de esquadrias em que o problema de corte de estoque bidimensional com baixa demanda foi resolvido.

Christofides e Hadjiconstantinou (1995) apresentaram um algoritmo de busca em árvore utilizando limitantes derivados de uma relaxação do espaço de estados de uma formulação de programação dinâmica, e um procedimento do tipo otimização do sub-gradiente.

Hifi (1997) apresentou melhorias no algoritmo proposto por Viswanathan and Bagchi (1993) enquanto que Cung *et al.* (2000) apresentaram uma nova versão do procedimento proposto por Hifi (1997). Lodi *et al.* (2002) realizaram uma revisão de proble-

mas de corte de estoque bidimensionais. Neste trabalho os autores discutem modelagens matemáticas, limitantes inferiores, algoritmos clássicos de aproximação, procedimentos heurísticos e metaheurísticos para resolver os problemas. Ainda em 2002, Alvarez-Valdés *et al.* desenvolveram vários procedimentos heurísticos para resolver o problema de corte bidimensional restritos e não restritos, com o objetivo de maximizar o valor dos itens cortados. Outros procedimentos heurísticos também foram desenvolvidos por Burke *et al.* (2004), Dowsland *et al.* (2006) e Cui (2008).

Hifi (2004) propôs um algoritmo para resolver o problema de corte bidimensional restrito. O algoritmo proposto é uma abordagem híbrida que combina a estratégia *hill-climbing* e algumas técnicas de programação dinâmica.

Recentemente, Morabito e Pureza (2008) desenvolveram um método heurístico para a geração de padrões de corte guilhotinados restritos. Neste trabalho, os autores combinam uma variante do método de Christofides e Hadjiconstantinou (1995) com a abordagem de busca em grafo E/OU.

Embora a literatura para problemas de corte de estoque unidimensionais e bidimensionais seja vasta, são poucos os trabalhos que consideram o aproveitamento de sobras. Na Seção 2.1 a seguir, apresentamos uma revisão mais detalhada dos trabalhos orientados para o aproveitamento de sobras no caso unidimensional. Para problemas de corte bidimensionais desconhecemos qualquer trabalho que considere o aproveitamento de sobras.

2.1 Revisão dos problemas de corte unidimensionais com sobras aproveitáveis

Os problemas de corte de estoque podem ser caracterizados por uma série de objetivos específicos e restrições, as quais não permitem a aplicação direta de modelos e algoritmos já desenvolvidos.

Nesta seção, focamos o problema de corte de estoque unidimensional com sobras de material aproveitáveis e apresentamos estudos da literatura que envolvem esses problemas. Para todos os trabalhos estudados fazemos uma padronização das notações utilizadas e,

desta forma, uma notação global para apresentar os modelos matemáticos foi proposta.

Para uma melhor organização do texto, os trabalhos foram ordenados conforme seus métodos de resolução (heurísticos e meta-heurísticos) e com ênfase em suas abordagens (orientações a padrões ou itens). No caso de problemas que apresentam uma abordagem híbrida, consideramos a abordagem utilizada para resolver a maior parte do problema de corte.

2.1.1 Notação Global

Índices

- i : referente ao item;
- j : referente ao padrão de corte;
- k : referente ao objeto.

Estoque

- K : número de tipos de objetos em estoque;
- L_k : comprimento do objeto tipo k ;
- e_k : disponibilidade do objeto tipo k ;
- c_k : custo unitário do objeto tipo k .

Demanda

- m : número de tipos de itens;
- l_i : comprimento do item tipo i ;
- d_i : demanda do item tipo i .

Parâmetros

- N_k : número de possíveis padrões de corte para o objeto tipo k ;

- δ_k : comprimento mínimo para uma sobra ser classificada como retalho no objeto k ;
- α_{ijk} : número de itens tipo i no padrão de corte j para o objeto tipo k .

Variáveis

- x_{jk} : número objetos tipo k cortados conforme o padrão de corte j (frequência);
- p_{ik} : número de itens tipo i cortados do objeto k ;
- $s_k = L_k - \sum_{i=1}^m \ell_i p_{ik}$: sobra no objeto k .

Esta notação foi proposta considerando que temos vários tipos de objetos disponíveis no estoque. Entretanto, se apenas um tipo de objeto é considerado nas formulações, o índice k é omitido.

Além desta notação proposta, outras variáveis e parâmetros podem ser definidos localmente em cada artigo apresentado, quando necessário.

2.1.2 Abordagens orientadas a itens

Nesta abordagem relacionamos os trabalhos em que todos os itens a serem cortados são tratados explicitamente, de modo que a maneira como um objeto é cortado é conhecida com a solução do problema. Assim, as variáveis fundamentais são o número de itens de um determinado tipo em um particular objeto, ou tipo de objeto.

- **Sinuary-Stern e Weiner (1994) - The one dimensional cutting stock problem using two objectives**

Neste trabalho Sinuary-Stern e Weiner consideram o problema de corte de estoque com dois objetivos a serem alcançados simultaneamente. O objetivo primário (mais importante) deveria minimizar a sobra gerada e o objetivo secundário, acumular a máxima quantidade de sobras no último objeto a ser cortado para que esta fosse maior que um determinado limitante (comprimento do menor item demandado) e retornasse ao estoque como retalho para ser utilizado no futuro. O algoritmo proposto foi desenvolvido para uma pequena oficina de Kibbutz Samar em Israel que trabalha com o corte de barras e

tubos de metal. O estoque da oficina era composto por objetos padronizados (adquiridos no mercado) e retalhos originados de cortes anteriores.

Além das variáveis já definidas, a formulação matemática desenvolvida pelos autores também considera:

- N : número de objetos usados.

Considerando inicialmente apenas objetos padronizados em estoque, o problema foi modelado como um problema linear inteiro e definido como:

$$\text{minimizar } N.L - \sum_{i=1}^m \ell_i d_i \quad (\text{equivalente a minimizar } N) \quad (2.1)$$

$$\text{maximizar } L - \sum_{i=1}^m \ell_i p_{iN} \quad (2.2)$$

sujeito a:

$$\sum_{i=1}^m \ell_i p_{ik} \leq L \quad k = 1, \dots, N \quad (2.3)$$

$$\sum_{k=1}^N p_{ik} \geq d_i \quad i = 1, \dots, m \quad (2.4)$$

$$p_{ik} \geq 0, \quad N \geq 0 \text{ e inteiro, } i = 1, \dots, m, \quad k = 1, \dots, N \quad (2.5)$$

Conforme relatado pelos autores, o algoritmo desenvolvido resolve problemas pequenos, sendo ineficiente para problemas grandes, além disso, o computador disponível na oficina era limitado e tinha uma pequena disponibilidade de softwares. Assim a estratégia desenvolvida foi resolver os problemas considerando o método de enumeração implícita (*branch and bound*), o qual iniciava com um limitante inferior simples para o número de barras:

$$N_0 = \left\lceil \frac{1}{L} \sum_{i=1}^m \ell_i d_i \right\rceil$$

em que $\lceil x \rceil$ é o menor inteiro maior ou igual a x .

Quando existia uma solução factível para as restrições do problema, então o problema (2.3)-(2.5) era resolvido considerando o segundo objetivo, caso contrário, havia um aumento no número de barras (1 de cada vez) até obter uma solução factível. Segundo os autores, com esse procedimento boas soluções foram encontradas para o problema.

- **Gradisar et al. (1997) - Optimization of roll cutting in clothing industry**

Neste artigo os autores apresentaram um estudo sobre o problema de corte unidimensional em indústrias de tecido, as quais tinham o estoque composto por objetos (*rolos*) com todos os comprimentos diferentes.

Para resolver este problema, que considera condições práticas específicas, os autores propuseram um modelo bi-objetivo para minimizar o número de itens que não são atendidos durante o processo de corte e a perda total (soma de perdas inferiores a um limitante). Este estudo também incluiu a possibilidade de retalhos retornarem para o estoque. O modelo matemático desenvolvido considera:

- Y : número máximo de tipos de itens que pode ser cortado de um objeto.

- z_k : indica se o objeto k é utilizado no plano de corte, $k = 1, \dots, K$:

$$z_k = \begin{cases} 0, & \text{se } p_{ik} = 0, \quad i = 1, \dots, m \\ 1, & \text{caso contrário.} \end{cases} \quad (2.6)$$

- y_{ik} : indica se o item i é cortado do objeto k , $i = 1, \dots, m, k = 1, \dots, K$:

$$y_{ik} = \begin{cases} 0, & \text{se } p_{ik} = 0 \\ 1, & \text{caso contrário.} \end{cases} \quad (2.7)$$

- t_k : indica o comprimento da sobra no objeto k , $k = 1, \dots, K$:

$$t_k = \begin{cases} s_k, & \text{se } z_k = 1 \text{ e } s_k \leq \delta \\ 0, & \text{caso contrário.} \end{cases} \quad (2.8)$$

Observe que o índice k foi omitido do parâmetro δ . Isto significa que o comprimento do retalho é o mesmo para qualquer objeto do estoque.

O modelo matemático proposto é dado por:

$$\text{minimizar } \sum_{i=1}^m \Delta_i \quad (2.9)$$

$$\text{minimizar } \sum_{k=1}^K t_k \quad (2.10)$$

sujeito a:

$$\sum_{i=1}^m \ell_i p_{ik} + s_k = L_k, \quad k = 1, \dots, K \quad (\text{restrições da mochila}) \quad (2.11)$$

$$\sum_{k=1}^K p_{ik} = d_i - \Delta_i, \quad i = 1, \dots, m \quad (\text{restrições de demanda}) \quad (2.12)$$

$$\sum_{i=1}^m y_{ik} \leq Y \leq M, \quad k = 1, \dots, K \quad (2.13)$$

$$p_{ik} \geq 0 \quad \text{e inteiro}, \quad i = 1, \dots, m, \quad k = 1, \dots, K \quad (2.14)$$

$$s_k \geq 0, \quad t_k \geq 0, \quad k = 1, \dots, K \quad (2.15)$$

$$\Delta_i \geq 0, \quad i = 1, \dots, m \quad (2.16)$$

$$y_{ik} \in \{0, 1\}, \quad z_k \in \{0, 1\}, \quad i = 1, \dots, m, \quad k = 1, \dots, K \quad (2.17)$$

No modelo (2.9)-(2.17), a função objetivo (2.9) minimiza o número de itens não cortados (a carência de itens é distribuída igualmente entre os tipos demandados para não prejudicar a produção de um determinado grupo de itens e conseqüentemente déficit na produção de alguns produtos) e a função objetivo (2.10) minimiza as perdas inferiores a um determinado limitante. Na restrição 2.13, os autores definiram $M = 4$. Este valor foi utilizado pelo fato de experimentos práticos mostrarem que quando $Y = 4$ o comprimento da perda tem uma redução não observada para qualquer outro valor de M .

O modelo proposto também faz distinção entre sobras que seriam descartadas (*perdas*) e sobras que retornariam ao estoque (*retalhos*). Neste trabalho o comprimento estabelecido para que uma sobra fosse considerada retalho (não contabilizado como perda) foi:

$$\delta \geq \min \{\ell_i, i = 1, \dots, m\}.$$

Embora tenham proposto o modelo (2.9)-(2.17), os autores não utilizaram este modelo para resolver o problema, talvez pela complexidade computacional e, desta forma, desenvolveram um procedimento heurístico, denominado COLA (COmputerized LAying out), o qual pode ser considerado como um algoritmo guloso. De acordo com a descrição do algoritmo COLA, os objetos em estoque são ordenados de modo não decrescente e, para cada objeto nesta seqüência, um padrão de corte é construído considerando três

possíveis formas de ordenação para os itens. Os padrões de corte são gerados utilizando o problema da mochila (para cada ordenação o valor de utilidade dos itens é alterado) e o padrão que apresentar a menor perda é utilizado.

Os testes computacionais realizados com problemas gerados aleatoriamente mostraram um bom desempenho do algoritmo COLA que, segundo os autores, gerou soluções em que a soma das perdas era inferior ao comprimento do menor item demandado.

- **Gradisar *et al.* (1999a) - A sequential heuristic procedure for one-dimensional cutting**

Neste artigo os autores desenvolveram um procedimento heurístico para resolver o problema de corte de estoque unidimensional em que o objetivo é a redução da perda quando todos os objetos em estoque possuem comprimentos diferentes e diferentes condições que podem aparecer na prática são consideradas, dentre elas, o aproveitamento de sobras. O método de solução proposto busca generalizar e melhorar o trabalho de Gradisar *et al.* (1997).

Devido as condições práticas que podem surgir em uma indústria durante o processo de corte, dois possíveis casos foram analisados e, para cada caso uma modelagem matemática foi proposta.

Nos modelos descritos a seguir, o parâmetro Y e as funções z_k , y_{ik} e t_k são as mesmas utilizadas em Gradisar *et al.* (1997). Além destas funções, os autores também definiram:

- u_k : indica se a sobra no objeto k é um retalho.

$$u_k = \begin{cases} 1, & \text{se } z_k = 1 \text{ e } s_k \geq \max \ell_i \\ 0, & \text{caso contrário.} \end{cases} \quad (2.18)$$

Caso 1: O estoque é suficiente para atender toda a demanda.

$$\text{minimizar } \sum_{k=1}^K t_k \quad (\text{minimiza a perda que deve ser inferior a } \delta) \quad (2.19)$$

sujeito a:

$$\sum_{i=1}^m \ell_i p_{ik} + s_k = L_k, \quad k = 1, \dots, K \quad (\text{restrições da mochila}) \quad (2.20)$$

$$\sum_{k=1}^K p_{ik} = d_i, \quad i = 1, \dots, m \quad (\text{restrições de demanda}) \quad (2.21)$$

$$\sum_{k=1}^K u_k \leq 1 \quad (\text{limita o número de retalhos}) \quad (2.22)$$

$$\sum_{i=1}^m y_{ik} \leq Y \leq m, \quad k = 1, \dots, K \quad (2.23)$$

$$p_{ik} \geq 0 \quad \text{e inteiro}, \quad i = 1, \dots, m, k = 1, \dots, K \quad (2.24)$$

$$s_k \geq 0, \quad t_k \geq 0, \quad k = 1, \dots, K. \quad (2.25)$$

Para o caso 1, como o estoque é suficiente para atender toda a demanda, o valor de δ pode ser definido como qualquer valor arbitrário entre 0 e $\max \{\ell_i, i = 1, \dots, m\}$.

Observe que este modelo é o mesmo proposto por Gradisar *et al.* (1997), porém, como existe material suficiente para atender a demanda, apenas a função objetivo (2.10) é considerada e a função u_k foi incluída para limitar o número máximo de retalhos produzidos.

Caso 2: O estoque é insuficiente para atender toda a demanda.

Neste caso, duas condições práticas foram analisadas na elaboração do modelo matemático.

- Condição 1: Não há preocupação com a distribuição dos itens no objeto a ser cortado.

$$\text{minimizar } \sum_{i=1}^m \Delta_i \ell_i \quad (2.26)$$

sujeito a:

$$\sum_{i=1}^m \ell_i p_{ik} + s_k = L_k, \quad k = 1, \dots, K \quad (2.27)$$

$$\sum_{k=1}^K p_{ik} = d_i - \Delta_i, \quad i = 1, \dots, m \quad (2.28)$$

$$\sum_{i=1}^m y_{ik} \leq Y \leq m, \quad k = 1, \dots, K \quad (2.29)$$

$$\Delta_i \geq 0, \quad i = 1, \dots, m. \quad (2.30)$$

No modelo (2.26)-(2.30) a função objetivo (2.26) minimiza o comprimento total de itens não cortados.

- Condição 2: Há preocupações com a distribuição dos itens no objeto a ser cortado.

Isso significa que alguns itens são mais importantes e, portanto, suas demandas devem ser atendidas primeiramente. Entretanto, pode haver conflito entre a perda e a distribuição desses itens que devem ter prioridade de corte. Desta forma, os autores consideram que se a perda é mais importante e a distribuição dos itens é aproximada, o objetivo (2.26) é alterado para:

$$\text{minimizar } \sum_{i=1}^m \Delta_i \text{ (minimiza o número de itens não cortados)} \quad (2.31)$$

$$\text{minimizar } \sum_{k=1}^K s_k \text{ (minimiza a perda que deve ser inferior a } \max \ell_i) \quad (2.32)$$

No caso em que a distribuição dos itens é mais importante que a perda, então a escolha de qualquer distribuição a priori é assumida. Os planos de corte são elaborados e o problema passa a ter um único objetivo a ser satisfeito, sendo que a perda mínima é determinada considerando a distribuição realizada. Desta forma, o objetivo (2.26) na condição 1 é alterado para:

$$\text{minimizar: } \sum_{i=1}^m \Delta_i \text{ (minimiza o número de itens não cortados)} \quad (2.33)$$

Assim como em Gradisar *et al.* (1997) os modelos propostos também não foram resolvidos e, desta forma, os autores desenvolveram um procedimento heurístico, denominado CUT o qual utiliza o Problema da Mochila (Gilmore e Gomory, 1963) com alterações que consideram o caso em que todos os objetos são de tamanhos diferentes.

Os passos utilizados para a construção do algoritmo CUT consiste em:

Passo 1: Selecionar os itens que serão cortados (os itens a serem produzidos são ordenados considerando seus comprimentos (ℓ_i) ou o número de itens não cortados (Δ_i));

Passo 2: Selecionar os objetos e gerar os padrões de corte (os objetos são ordenados de modo crescente);

Passo 3: Se existir objetos em estoque para serem cortados e demanda de itens para ser atendida volte para o passo 1, caso contrário, pare.

Segundo os autores, CUT é uma versão melhorada e generalizada do algoritmo COLA (Gradisar *et al.* (1997)) e encontra soluções melhores em um tempo computacional muito baixo.

- **Abuabara e Morabito (2009): Cutting optimization of structural tubes to build agricultural light aircrafts**

Recentemente, Abuabara e Morabito aplicaram o problema de corte de estoque com sobras de material aproveitáveis em uma empresa brasileira que tem em sua linha de produção o corte de tubos estruturais metálicos para produção de aeronaves agrícolas.

O problema a ser resolvido considera uma lista de diferentes tipos de objetos em estoque em quantidades suficientes para atender toda demanda de itens e possíveis retalhos que foram gerados em cortes anteriores.

Para resolver o problema, os autores propuseram dois modelos matemáticos, os quais foram resolvidos utilizando um software comercial. O primeiro modelo (Modelo 1) corresponde ao modelo de Gradisar *et al.* (1997), porém reescrito como um problema inteiro misto (PIM) e, o segundo (Modelo 2) consiste em simplificações realizadas no modelo anterior.

Para reescrever as funções (2.6) e (2.8) os autores utilizam no modelo (2.34)-(2.50) a seguir, o parâmetro M como sendo um número suficientemente grande e w_k como uma variável binária que indica quando $s_k < \delta$ ($w_k = 1$) ou $s_k \geq \delta$ ($w_k = 0$). Além de M e w_k , o parâmetro ε é utilizado como um número positivo pequeno para mudar o sinal das desigualdades de “ \geq ” para “ $>$ ”.

Além das funções (2.6) e (2.8), os autores também incluíram no modelo a função (2.18) que indica se a sobra no objeto k é um retalho. Para utilizar esta função no modelo, a variável binária w_k também foi utilizada para indicar se $s_k \geq \delta$ ($w_k = 0$) ou $s_k < \delta$ ($w_k = 1$).

O valor de δ é definido entre $\min \{\ell_i, i = 1, \dots, m\}$ e $\min \{L_k, k = 1, \dots, K\}$. Qualquer valor inferior a δ seria considerado perda e conseqüentemente seria descartado. O modelo de Abuabara e Morabito não limita o número de diferentes tipos de itens que devem compor o padrão de corte (restrição 2.13) e, como existe estoque suficiente para atender toda a demanda de itens, a função objetivo 2.9 não é utilizada neste modelo.

Modelo 1

$$\text{minimizar } \sum_{k=1}^K t_k \quad (\text{minimiza a perda}) \quad (2.34)$$

sujeito a:

$$\sum_{i=1}^m \ell_i p_{ik} + s_k = L_k, \quad k = 1, \dots, K \quad (\text{restrições da mochila}) \quad (2.35)$$

$$\sum_{k=1}^K p_{ik} = d_i, \quad i = 1, \dots, m \quad (\text{restrições de demanda}) \quad (2.36)$$

$$z_k \leq \sum_{i=1}^m p_{ik}, \quad k = 1, \dots, K \quad (2.37)$$

$$\sum_{i=1}^m p_{ik} \leq M z_k, \quad k = 1, \dots, K \quad (2.38)$$

$$(s_k - N) \geq -M w_k + \varepsilon, \quad k = 1, \dots, K \quad (2.39)$$

$$(s_k - N) \leq M(1 - w_k), \quad k = 1, \dots, K \quad (2.40)$$

$$t_k - M w_k \leq 0, \quad k = 1, \dots, K \quad (2.41)$$

$$t_k - M z_k \leq 0, \quad k = 1, \dots, K \quad (2.42)$$

$$-s_k + t_k \leq 0, \quad k = 1, \dots, K \quad (2.43)$$

$$s_k - t_k + M w_k + M z_k \leq 2M, \quad k = 1, \dots, K \quad (2.44)$$

$$-z_k + u_k \leq 0, \quad k = 1, \dots, K \quad (2.45)$$

$$w_k + u_k \leq 1, \quad k = 1, \dots, K \quad (2.46)$$

$$z_k - w_k - u_k \leq 0, \quad k = 1, \dots, K \quad (2.47)$$

$$\sum_{k=1}^K u_k \leq 1 \quad (\text{limita a quantidade de retalhos}) \quad (2.48)$$

$$p_{ik} \geq 0 \text{ e inteiro}, \quad s_k \geq 0, \quad t_k \geq 0, \quad i = 1, \dots, m, \quad k = 1, \dots, K \quad (2.49)$$

$$z_k \in \{0, 1\}, \quad w_k \in \{0, 1\}, \quad u_k \in \{0, 1\}, \quad k = 1, \dots, K. \quad (2.50)$$

Na função objetivo (2.34) deste modelo, $t_j = s_k$ se $s_k \leq \delta$ ou $t_j = 0$, caso contrário.

Note que o Modelo 1 pode não apresentar soluções factíveis mesmo que a demanda de itens seja atendida. Isto ocorre caso a restrição (2.48) seja violada, ou seja, dois ou mais retalhos sejam gerados durante o processo de corte. Entretanto, essa falha não impediu o Modelo 1 de encontrar soluções para os testes computacionais realizados.

Modelo 2

$$\text{minimizar } \sum_{k=1}^K t_k \quad (\text{minimiza a perda}) \quad (2.51)$$

sujeito a:

$$\sum_{i=1}^m \ell_i p_{ik} \leq L_k, \quad k = 1, \dots, K \quad (\text{restrições da mochila}) \quad (2.52)$$

$$\sum_{k=1}^K p_{ik} = d_i, \quad i = 1, \dots, m \quad (\text{restrições de demanda}) \quad (2.53)$$

$$Nu_k \leq L_k z_k - \sum_{i=1}^m \ell_i p_{ik}, \quad k = 1, \dots, K \quad (2.54)$$

$$L_k z_k - \sum_{i=1}^m \ell_i p_{ik} \leq t_k + u_k M, \quad k = 1, \dots, K \quad (2.55)$$

$$\sum_{k=1}^K u_k \leq 1 \quad (\text{limita a quantidade de retalhos}) \quad (2.56)$$

$$p_{ik} \geq 0 \text{ e inteiro}, \quad t_k \geq 0, \quad i = 1, \dots, m, \quad k = 1, \dots, K \quad (2.57)$$

$$z_k \in \{0, 1\}, \quad u_k \in \{0, 1\}, \quad k = 1, \dots, K. \quad (2.58)$$

Observe que uma ligeira modificação nos modelos 1 e 2 pode considerar o critério de minimizar o comprimento total cortado de objetos. Para soluções alternativas de perda mínima, a idéia seria favorecer aquelas que evitam a geração de novos retalhos. Neste caso, a função objetivo (2.51) deveria ser substituída por:

$$\text{minimizar } \sum_{k=1}^K t_k + \frac{\sum_{k=1}^K L_k z_k}{\sum_{k=1}^K L_k} \quad (2.59)$$

Com algumas alterações nos modelos 1 e 2, os autores também apresentaram a possibilidade de planejar a produção em multi-períodos, levando em consideração os benefícios de antecipar a produção de alguns itens e mantê-los em estoque ao invés de gerar padrões de corte com retalhos para o uso no atendimento de futuras demandas.

Os testes computacionais realizados considerando o objetivo de minimizar a perda gerada, mostrou um melhor desempenho do Modelo 2 em relação ao Modelo 1 e as soluções apresentadas pela empresa. Outros resultados apresentados mostram o trade-off entre a perda total e o número máximo de retalhos permitidos durante o processo de corte.

- **Gradisar e Trkman (2005) - A combined approach to the solution to the general one-dimensional cutting stock problem**

Com a finalidade de mostrar que os métodos existentes para resolver o problema de corte podem ser melhorados fazendo uma combinação de métodos exatos e procedimentos heurísticos, nesse trabalho os autores desenvolveram uma abordagem na qual combinam o procedimento heurístico CUT (Gradisar *et al.* (1999a)), que resolve a maior parte do problema e o método *branch and bound* que resolve o problema residual final. O objetivo da combinação desses dois métodos é obter perda mínima em um tempo computacional aceitável.

Uma característica desse problema é que todos os objetos em estoque possuem comprimentos diferentes e ao final do processo de corte, as sobras geradas, se suficientemente grandes, são aproveitadas no futuro e não são contabilizadas como perda. O algoritmo proposto foi denominado de C-CUT (Combined Cutting).

Nos modelos descritos a seguir, as funções t_k e u_k são as mesmas definidas em Gradisar *et al.* (1999a). Além destas funções, os autores definiram:

- y_k : indica se o objeto k é utilizado no plano de corte;

$$y_k = \begin{cases} 0, & \text{se o objeto } k \text{ é utilizado no plano de corte} \\ 1, & \text{caso contrário.} \end{cases} \quad (2.60)$$

Como os autores consideram condições práticas que podem surgir em uma indústria durante o processo de corte, dois possíveis casos foram analisados e, para cada caso uma modelagem matemática foi proposta.

Caso 1: O estoque é suficiente para atender toda a demanda.

$$\text{minimizar } \sum_{k=1}^K t_k \text{ (minimiza a perda)} \quad (2.61)$$

sujeito a:

$$\sum_{i=1}^m \ell_i p_{ik} + s_k = L_k(1 - y_k), \quad k = 1, \dots, K \text{ (restrições da mochila)} \quad (2.62)$$

$$\sum_{k=1}^K p_{ik} = d_i, \quad i = 1, \dots, m \text{ (restrições de demanda)} \quad (2.63)$$

$$s_k - \delta u_k \geq 0, \quad k = 1, \dots, K \quad (2.64)$$

$$\sum_{k=1}^K u_k \leq 1 \quad (2.65)$$

$$p_{ik} \geq 0 \text{ e inteiro, } i = 1, \dots, m, k = 1, \dots, K \quad (2.66)$$

$$s_k \geq 0, t_k \geq 0, \quad k = 1, \dots, K \quad (2.67)$$

$$u_k \in \{0, 1\}, y_k \in \{0, 1\}, \quad k = 1, \dots, K. \quad (2.68)$$

Como existe estoque suficiente para atender toda a demanda, o valor de δ pode ser definido como qualquer valor arbitrário entre 0 e $\max \{\ell_i, i = 1, \dots, m\}$. Além disso, a restrição (2.65) garante que apenas uma sobra seja computada como retalho, caso contrário, o modelo (2.61)-(2.68) poderia gerar padrões de corte sem perdas, porém com muitos retalhos.

Caso 2: O estoque é insuficiente para atender toda a demanda e a distribuição dos itens a serem cortados não é importante.

$$\text{minimizar } \sum_{k=1}^K s_k \text{ (minimiza a soma das sobras)} \quad (2.69)$$

sujeito a:

$$\sum_{i=1}^m \ell_i p_{ik} + s_k = L_k, \quad k = 1, \dots, K \quad (2.70)$$

$$\sum_{k=1}^K p_{ik} \leq d_i, \quad i = 1, \dots, m \quad (2.71)$$

$$p_{ik} \geq 0 \text{ e inteiro, } s_k \geq 0, \quad i = 1, \dots, m, k = 1, \dots, K \quad (2.72)$$

A idéia básica do C-CUT é encontrar uma solução temporária com o algoritmo CUT e então melhorá-la resolvendo um problema menor (sub-problema) com o método exato. O sub-problema é gerado a partir da solução obtida pelo procedimento CUT e considera os casos em que existe quantidade suficiente de objetos em estoque para atender a demanda (caso 1) e quando há carência de material (caso 2).

Os passos utilizados para a construção do algoritmo C-CUT consiste em:

Passo 1: Obter uma solução com o procedimento CUT e armazená-la;

Passo 2: Se a solução for ótima, PARE. A solução é considerada ótima se:

- $\sum_{k=1}^K t_k = 0$ (caso 1).
- $\sum_{k=1}^K s_k = 0$ (caso 2).
- $0 \leq \left(\sum_{k=1}^K L_k - \sum_{i=1}^m \ell_i d_i \right) < \delta$ e $\sum_{k=1}^K t_k = \sum_{k=1}^K L_k - \sum_{i=1}^m \ell_i d_i$.

Passo 3: Gerar o sub-problema a partir dos resultados obtidos no Passo 1.

- Caso 1: desmontar os padrões gerados pelo algoritmo CUT em que $s_k \neq 0$. Neste caso, uma demanda residual é gerada e o estoque é atualizado com os objetos dos padrões de corte desmontados;
- Caso 2: desmontar os padrões gerados pelo algoritmo CUT em que $s_k \neq 0$. Neste caso, o estoque residual é composto apenas por objetos dos padrões de corte desmontados e a demanda residual possui itens dos padrões de corte desmontados e itens que não foram produzidos devido a falta de material.

Passo 4: Resolver o sub-problema formado pelos padrões de corte desmontados no Passo 3 utilizando o método *branch and bound*;

Passo 5: Agrupar os padrões de corte não desmontados (Passo 1) com a solução obtida pelo sub-problema. Comparar esta solução com a solução do algoritmo CUT e selecionar a melhor.

Os testes computacionais realizados mostraram superioridade da solução do algoritmo C-CUT em relação ao algoritmo CUT.

- **Trkman and Gradisar (2007) - One-dimensional cutting stock optimization in consecutive time periods**

O estudo desenvolvido por Trkman e Gradisar (2007) considera o problema de corte de estoque unidimensional em que consecutivos períodos devem ser resolvidos. Neste trabalho que tem como objetivo a minimização da perda ou dos custos sobre um determinado período de tempo, os autores consideram que todos os objetos em estoque possuem complementos diferentes e consideram a possibilidade de sobras retornarem ao estoque desde

que sejam superiores a um determinado limitante. Os itens são cortados para estoque ou para clientes, entretanto, o comprimento e a demanda desses itens não são conhecidos a priori.

O método proposto para resolver o problema é satisfatório no caso em que os custos do material (objetos) e das perdas são relativamente altos quando comparados com outros custos e quando há uma grande economia em aproveitar retalhos. Como os autores consideram consecutivos períodos, o retalho gerado em um determinado período é armazenado para ser cortado no período seguinte. Além disso, a cada período, uma determinada quantidade de novos objetos é adicionada aos que já estão disponíveis no estoque. Este problema de corte de estoque multi-período, de fato, não é tratado conforme Poldi e Arenales (2007), em que a demanda dos itens ocorre nos diversos períodos de um horizonte de planejamento.

Inicialmente os autores propuseram um modelo matemático que apresentou boas soluções apenas quando um período era considerado, entretanto, boas soluções em um período não levam a otimização ao longo de vários períodos e, desta forma, o modelo não apresentou soluções aceitáveis quando vários períodos eram considerados. Muitos problemas tornavam-se infactíveis durante os períodos, havia um aumento na perda e muitos retalhos retornavam ao estoque gerando custos adicionais para a empresa como custos com transporte e administração dos retalhos.

O segundo modelo proposto, baseado no anterior, tinha como objetivo a minimização das perdas e incluía custos de retornar retalhos para estoque. Com esta condição, uma baixa quantidade de retalhos retornaria ao estoque, mesmo que isso significasse um ligeiro aumento na perda do período corrente. O modelo matemático desenvolvido tem como objetivo a minimização de custos e considera os seguintes dados:

- c : custo de uma unidade de material perdida;
- p_r : custo de retornar um retalho para o estoque;
- g_k : representa os retalhos que retornam para o estoque

$$g_k = \begin{cases} 1, & \text{se e somente se } s_k \geq \delta \text{ e } (cs_k) > p_r \\ 0, & \text{caso contrário.} \end{cases} \quad (2.73)$$

As funções t_k e y_k presentes no modelo a seguir são as mesmas consideradas em Gradisar and Trkman (2005) e o comprimento para o retalho é dado por $\{\min L_k \geq \delta \geq \min \ell_i, k = 1, \dots, K, i = 1, \dots, m\}$.

$$\text{minimizar } \sum_{k=1}^K (t_k c + g_k p_r) \quad (2.74)$$

sujeito a:

$$\sum_{i=1}^m \ell_i p_{ik} + s_k = L_k(1 - y_k), \quad k = 1, \dots, K \quad (\text{restrições da mochila}) \quad (2.75)$$

$$\sum_{k=1}^K p_{ik} = d_i, \quad i = 1, \dots, m \quad (\text{restrições de demanda}) \quad (2.76)$$

$$\delta - s_k + \delta(g_k - 1) \leq 0 \quad (2.77)$$

$$s_k - t_k - (g_k + y_k)(\max L_k) \leq 0, \quad k = 1, \dots, K \quad (2.78)$$

$$p_{ik} \geq 0 \text{ e inteiro, } s_k \geq 0, \quad t_k \geq 0, \quad i = 1, \dots, m, \quad k = 1, \dots, K \quad (2.79)$$

$$g_k \in \{0, 1\}, \quad y_k \in \{0, 1\}, \quad k = 1, \dots, K. \quad (2.80)$$

Segundo os autores, a solução deste modelo apresenta uma perda baixa, utiliza poucos objetos do estoque e retorna uma baixa quantidade de retalhos. Isto faz com que a empresa tenha baixo custo com o transporte destes objetos.

2.1.3 Abordagens orientadas a padrões

Nesta abordagem relacionamos os trabalhos nos quais os itens demandados são arranjados em padrões de corte e frequências (número de objetos cortados segundo um padrão de corte) são determinadas para que a demanda seja atendida.

- **Roodman (1986) - Near-optimal solutions to one-dimensional cutting stock problems**

Este artigo descreve um procedimento heurístico para a geração de padrões para o problema de corte de estoque unidimensional na qual os objetos em estoque possuem comprimentos diferentes e têm disponibilidades limitadas. O objetivo principal é a minimização das perdas, entretanto existe um objetivo secundário que é a possibilidade

concentrar as sobras em poucos padrões de corte para que possam ser utilizadas no corte de futuros itens demandados. O trabalho foi desenvolvido para uma indústria de corte de aço que tem em sua linha de produção o corte de alguns itens grandes.

O autor desenvolveu o trabalho focando o objetivo de perda mínima enquanto que as sobras se refletiam na maneira em que o procedimento heurístico resolvia o problema para manter a factibilidade, pois alguns objetos tinham poucas disponibilidades em estoque. O problema de corte resolvido é o mesmo proposto por Gilmore e Gomory (1963):

$$P = \text{minimizar } \sum_j L_{k(j)} x_j \quad (2.81)$$

sujeito a:

$$\sum_j \alpha_{ij} x_j \geq d_i, \quad i = 1 \dots, m \quad (2.82)$$

$$\sum_{j \in V_k} x_j \leq e_k, \quad k = 1 \dots, K \quad (2.83)$$

$$x_j \geq 0 \text{ e inteiro.} \quad (2.84)$$

No modelo (2.81)-(2.84), α_{ij} é a quantidade de itens do tipo i cortados do padrão j , $k(j)$ é o índice do objeto que o padrão de corte j foi gerado e $V_k = \{j/k(j) = k\}$.

O procedimento heurístico desenvolvido para resolver o problema consiste de três fases. Na primeira fase, a solução inicial para os problemas é obtida pela relaxação do modelo (2.81)-(2.84) utilizando uma técnica modificada de geração de colunas seguida de um arredondamento que resulta em soluções inteiras. O arredondamento é realizado considerando que as soluções não são arredondadas para o inteiro superior ao fracionário obtido se alguma restrição for violada. A segunda fase do procedimento heurístico consiste em factibilizar o problema. Os itens que estão em excesso são considerados como sobra e desta forma é possível fazer uma reorganização nos itens que compõem os padrões de modo que as sobras fiquem concentradas em poucos padrões de corte. Estas sobras são utilizadas para cortar os itens que não tiveram suas demandas atendidas na primeira fase. Caso não seja possível atender toda demanda com as sobras geradas, um novo objeto do estoque é utilizado ou uma nova reorganização dos itens que compõem os padrões de corte é realizada.

A terceira fase consiste em fazer melhorias na solução obtida trocando padrões de corte que utilizam longos objetos do estoque por objetos com comprimentos menores ou vice-versa. Após essas mudanças, as sobras nos padrões de corte devem ser reorganizadas. Os testes computacionais foram realizados considerando 12 problemas testes sendo que alguns forneceram soluções ótimas (a solução da heurística foi comparada com a solução ótima de um PL) e outros, soluções muito próximas da otimalidade.

- **Gradisar *et al.* (1999b) - A hybrid approach for optimization of one-dimensional cutting**

Neste trabalho os autores propuseram uma abordagem que combina programação linear e procedimento heurístico (baseado em alterações no algoritmo COLA, Gradisar *et al.* (1997)) para encontrar a solução do problema de corte unidimensional. Além de atender a demanda dos itens nas quantidades solicitadas, a abordagem busca também acumular sobras em um único objeto cortado para que possam ser utilizadas no futuro.

Neste problema, diferentemente dos propostos em Gradisar *et al.* (1997) e Gradisar *et al.* (1999a), a maior parte do estoque é composta por objetos com o mesmo comprimento ou então por poucos objetos com comprimentos diferentes. Somente uma pequena parte do estoque pode ter todos os comprimentos diferentes que são retalhos de cortes anteriores que retornaram ao estoque para atender futuras demandas.

Para resolver este problema, os autores propõe um método de solução composto por 2 estágios. No primeiro estágio, o problema é modelado e resolvido como um problema linear (S_{LPM}) considerando apenas os objetos padronizados (objetos longos e com repetição). Seja \bar{k} o número de tipos de objetos padronizados em estoque:

$$\text{minimizar } \sum_{k=1}^{\bar{k}} \sum_{j=1}^{N_k} L_k x_{jk} \quad (2.85)$$

sujeito a:

$$\sum_{k=1}^{\bar{k}} \sum_{j=1}^{N_k} \alpha_{ijk} x_{jk} \geq d_i, \quad i = 1, \dots, m \quad (2.86)$$

$$x_{jk} \geq 0 \text{ e inteiro, } j = 1 \dots, N_k, \quad k = 1 \dots, K \quad (2.87)$$

O problema (2.85) - (2.87) pode ser resolvido utilizando o método de programação proposto por Gilmore e Gomory (1963) ou qualquer outro método de programação linear.

Após resolver este problema, no segundo estágio, as frequências dos padrões de corte que geram demandas em excesso são reduzidas (uma a uma). A seleção dos padrões de corte que têm a frequência reduzida considera os padrões gerados pelos objetos mais longos. Com essa remoção muitas demandas de itens podem deixar de serem atendidas e, desta forma, estoque e demandas são atualizados e este problema residual é resolvido pelo procedimento heurístico (S_{SHP}) COLA (Gradisar *et al.* (1997)) com uma pequena alteração que distingue os objetos padronizados (objetos comprados pela empresa) e não padronizados (retalhos) do estoque e permite que apenas um retalho seja gerado (Gradisar *et al.* 1999a). O estoque de objetos disponíveis para atender a demanda residual é formado por um objeto padronizado de cada tipo e todos os retalhos do estoque.

A solução final (S), com a quantidade exata de itens demandados e sobras acumuladas em um único objeto é dada pela união das soluções obtidas:

$$S = (S_{LPM}) + (S_{SHP})$$

Neste trabalho, os autores apresentaram apenas um exemplo para ilustrar o desenvolvimento do método proposto e não fizeram qualquer comentário sobre possíveis testes realizados.

- **Scheithauer (1991) - A note on handling residual lengths**

Neste trabalho o autor propõe uma estratégia para controlar retalhos gerados (chamados pelo autor de ‘comprimentos residuais’) durante o processo de corte de estoque. O objetivo do trabalho é mostrar como retalhos podem ser considerados utilizando a técnica de geração de colunas, por definir previamente possíveis tamanhos de retalhos e usar o modelo de Gilmore e Gomore (1963).

A estratégia é restrita a problemas de corte unidimensionais com vários tipos de objetos L_k em estoque com custos v_k , $k = 1, \dots, K$ em quantidades ilimitadas. Os itens a serem produzidos têm comprimentos ℓ_i e demanda d_i , $i = 1, \dots, m$. Adicionalmente, alguns retalhos R_q com comprimentos r_q têm um valor positivo estimado w_q , $q = 1, \dots, p$. Encontrar as frequências de um conjunto de padrões de corte tal que as restrições de demanda sejam satisfeitas com um custo mínimo (o custo dos objetos utilizados menos o valor dos comprimentos residuais obtidos) é o objetivo desejado.

Para a formulação matemática do problema, a seguinte variável foi definida:

- $a_{jk} = (\alpha_{1jk}, \alpha_{2jk}, \dots, \alpha_{(m+p)jk})^T$: padrão de corte j para o objeto tipo k em estoque, $j = 1, \dots, N_k, k = 1, \dots, K$;

$$z = \text{minimizar} \sum_{k=1}^K \sum_{j=1}^{N_k} \left(v_k - \sum_{q=1}^p w_q \alpha_{(m+q)jk} \right) x_{jk} \quad (2.88)$$

sujeito a:

$$\sum_{k=1}^K \sum_{j=1}^{N_k} \alpha_{ijk} x_{jk} \geq d_i \quad i = 1, \dots, m \quad (2.89)$$

$$x_{jk} \geq 0 \text{ e inteiro, } j = 1, \dots, N_k, k = 1, \dots, K \quad (2.90)$$

Seja $\pi = (\pi_1, \pi_2, \dots, \pi_m)^T$ o vetor de multiplicadores simplex. O sub-problema a ser resolvido é dado por:

$$z_q = \text{minimizar} \left\{ v_k - \sum_{q=1}^p w_q \alpha_{(m+q)k} - \sum_{i=1}^m \pi_i \alpha_{ik} \text{ tal que} \right.$$

$$\left. \sum_{i=1}^m \ell_i \alpha_{ik} + \sum_{q=1}^p r_q \alpha_{(m+q)k} \leq L_k, \alpha_{ik} \geq 0, \text{ inteiro, } i = 1, \dots, m + p \right\}, k = 1, \dots, K.$$

Devido ao número exponencial de variáveis, a relaxação linear do modelo (2.88)-(2.90) foi resolvida utilizando a técnica de geração de colunas, de maneira análoga a Gilmore e Gomory (1963). O artigo fornece detalhes sobre o desenvolvimento do método assim como sugestões para estimativas dos retalhos.

Para gerar uma solução inteira para a solução ótima do problema relaxado, um procedimento de arredondamento foi adaptado (não publicado na língua inglesa) o qual também considera os comprimentos residuais.

Foram apresentados dois exemplos para ilustrar a estratégia desenvolvida. Estes exemplos apontam a dificuldade em considerar retalhos e a melhora que pode haver na solução quando são considerados. Para finalizar, o autor observa a facilidade de extensão para o problema de corte bidimensional guilhotinado.

- **Chu e Antonio (1999) - Approximation algorithms to solve real-life multi-criteria cutting stock problems**

O trabalho desenvolvido por Chu e Antonio considera um problema de corte unidimensional real em uma indústria especializada em cortar tubos de metal com vários perfis (circulares, quadrangulares, triangulares etc). O objetivo é minimizar os custos com a perda, tempo de corte necessário para a produção das peças e os possíveis retalhos que podem ser gerados durante o processo de corte.

Basicamente, o problema consiste em produzir m tipos de itens com comprimento ℓ_i e demanda d_i a partir do corte de K tipos de objetos com comprimento L_k e disponibilidade e_k . Para resolver este problema, os autores consideram um problema de corte unidimensional típico com algumas restrições técnicas que surgem na indústria de corte de metal.

- Tolerância

A tolerância está relacionada ao objeto a ser cortado. Existem duas categorias para a tolerância: *estrita* e *normal*. Itens com tolerância estrita são entregues ao cliente com comprimento exato e em boa qualidade (sem deformações geométricas e composição química uniforme). Em itens com tolerância normal, estas exigências não são consideradas.

- Ajuste

O conceito de ajuste está diretamente relacionado à tolerância. Nas extremidades do objeto a ser cortado pode existir deformações geométricas ou composição química heterogênea resultante do tratamento térmico. Desta forma, se um item estrito for alocado em uma das extremidades do objeto que apresente deformações, um determinado comprimento a da extremidade deve ser excluído. Um item com tolerância normal pode ser alocado em qualquer extremidade sem que um comprimento adicional seja excluído.

- Lotes

Durante o processo de corte, mais que um tipo de objeto pode ser cortado simultaneamente. Desta forma, a quantidade de cada tipo de item produzido é um múltiplo do número de objetos no lote (objetos cortados), sendo que o tempo requerido para cortar

um lote é menor que o tempo para cortar objetos individuais. O tempo necessário para executar um simples corte (custo) é conhecido para cada tamanho de lote.

Alguns tamanhos de lotes podem ser inactíveis, por razões técnicas (estabilidade, capacidade). Os custos de corte para tamanhos factíveis s devem satisfazer a seguinte relação:

$$p(s) \leq p(s - s') + p(s'), \quad 1 \leq s' < s$$

e $(s - s')$ e s' são tamanhos factíveis. Esta relação garante que cortar um lote grande sempre custa menos que cortar dois lotes menores. A preparação de um lote tem custo π independente do seu tamanho. A soma dos custos de corte e de preparação de um lote com tamanho s que requer d cortes é dada por:

$$\mu(s, d) = \pi + d \times p(s).$$

Devido à inactibilidade de alguns tamanhos de lotes, os autores definiram “lotes fictícios”, que são compostos de vários lotes com tamanhos factíveis. Para cada lote fictício, existem vários modos de decomposição e apenas uma decomposição de custo mínimo é considerada. Portanto, para um lote fictício com tamanhos inactíveis s , tem-se:

$$\mu(s, d) = \min_{1 \leq s' \leq \lfloor s/2 \rfloor} \{ \mu(s - s', d) + \mu(s', d) \}$$

em que $\lfloor x \rfloor$ denota a parte inteira de x .

- Material Reutilizável

Quando um objeto cortado possui uma sobra com comprimento maior que um determinado comprimento δ , esta pode voltar ao estoque para ser utilizada no atendimento de futuras demandas e desta forma, não ser contabilizada como perda. Entretanto, se não houver uma penalidade para a parte de material que retorna ao estoque, isto pode levar a situações com planos de corte com muitos retalhos. Para evitar esta situação, os seguintes critérios foram adotados: o custo de transporte dos retalhos é alto, os retalhos são estocados em uma área com restrição de capacidade e como todo retalho será cortado, estima-se para ele uma perda.

- Formulação Matemática

Para computar os custos de cada padrão de corte, os autores definiram e mostraram como computar funções para obter a perda $w(z, a)$, o comprimento do retalho $t(z, a)$ e o número de cortes requeridos $d(z, a)$ quando um subconjunto a de itens é alocado em um objeto com comprimento z . Além destas funções, os autores também definiram $\rho(z, a)$ que representa o custo com a perda e o retalho gerado quando um subconjunto a de itens é alocado em um objeto de comprimento z .

Considerando que existe N lotes, para cada lote j , $j = 1, \dots, N$, x_j é o tamanho do lote (frequência) e a_j é o subconjunto de itens alocados em cada objeto do lote j , a seguinte variável de decisão foi definida:

$$Z_j = \begin{cases} 1, & \text{se o objeto tipo } k \text{ é utilizado no pacote} \\ 0, & \text{caso contrário} \end{cases}$$

Seja z_{kj} o objeto tipo k utilizado no lote j . A formulação matemática do problema é dada por:

$$\text{maximizar: } \sum_{j=1}^N \{x_j \rho(Z_j, a_j) + \mu(x_j, d(Z_j, a_j))\} \quad (2.91)$$

sujeito a:

$$\sum_{j=1}^N \alpha_{ij} x_j = d_i \quad i = 1, \dots, m \quad (\text{restrições de demanda}) \quad (2.92)$$

$$\sum_{j=1}^N z_{kj} x_j \leq e_k \quad k = 1, \dots, m \quad (\text{restrições de estoque}) \quad (2.93)$$

$$\sum_{k=1}^K z_{kj} = 1 \quad j = 1, \dots, N \quad (2.94)$$

$$z_{kj} \in \{0, 1\}, \quad k = 1, \dots, K, \quad j = 1, \dots, N \quad (2.95)$$

$$x_j \geq 0 \quad j = 1, \dots, N \quad (2.96)$$

$$\alpha_{ij} \geq 0 \quad i = 1, \dots, m, \quad j = 1, \dots, N \quad (2.97)$$

Na função objetivo (2.91), o primeiro termo corresponde ao custo com a perda e retalhos e o segundo termo é referente ao custo de corte. A restrição (2.94) garante que um único tipo de objeto seja utilizado em cada lote.

Como a função objetivo do modelo (2.91) não é uma função linear das variáveis de decisão, um algoritmo de programação linear não é apropriado para resolver este problema. Desta forma, os autores mostram que é possível resolvê-lo na otimalidade utilizando programação dinâmica, porém, devido a complexidade computacional, sua utilização é impossível quando problemas reais são considerados. Desta forma, modificações foram propostas neste algoritmo sendo que soluções muito satisfatórias foram obtidas (os custos reduziram mais que 8% comparado com as soluções obtidas pela indústria) em um tempo aceitável. Para problemas gerados aleatoriamente o algoritmo proposto também apresentou boas soluções.

- **Cherri *et al.* (2009) - The one dimensional cutting stock problems with usable leftover: A heuristic approach**

Neste trabalho Cherri *et al.* consideram o problema de corte unidimensional em que as sobras geradas nos padrões de corte podem ser estocadas desde que tenham dimensões significativas. Estas sobras (denominadas de *retalhos*) não são contabilizadas como perdas e retornam ao estoque para atender futuras demandas. Para resolver este problema, alterações foram realizadas em procedimentos heurísticos (construtivos e residuais) clássicos da literatura.

O problema estudado pelos autores consiste em obter uma solução para o problema a partir do corte de objetos padronizados (objetos comprados pela empresa) e não padronizados (retalhos de corte anteriores) de modo que a demanda dos itens seja atendida e as sobras geradas pelos padrões de corte sejam aceitáveis para serem descartadas (perdas) ou suficientemente grandes para retornarem ao estoque (retalhos).

Alguns critérios e parâmetros para análise das soluções foram estabelecidos, assim como uma definição para classificar o procedimento heurístico com melhor desempenho.

Os procedimentos heurísticos construtivos para resolver o problema de aproveitamento de sobras, FFD_A e $Guloso_A$, foram desenvolvidos a partir dos procedimentos clássicos FFD e Guloso com a finalidade de gerar padrões de corte com perdas pequenas ou com retalhos para estoque. Estes dois procedimentos heurísticos são descritos a seguir.

- Procedimento FFD_A

O procedimento FFD_A consiste em aplicar o procedimento FFD (alocação de itens em ordem decrescente) para obter um padrão de corte para cada objeto disponível no estoque e, logo após, a sobra é analisada. Se a sobra for aceitável, isto é, pequena ou grande o suficiente para ser considerada retalho, o padrão é aceito. Senão um item do padrão (o maior) é retirado. Assim, para o espaço gerado é resolvido um problema da mochila, cuja capacidade é a sobra no padrão adicionada ao tamanho do item retirado. Depois de resolvido o problema da mochila, a sobra gerada é analisada e se não for aceitável, outro item do padrão inicial (segundo maior) é retirado. Novamente para o espaço gerado é resolvido o problema da mochila. Caso tenha sido retirado um item de cada comprimento dentre todos que compõem o padrão, volta-se retirar o primeiro maior. Este procedimento é repetido até que a sobra seja aceitável, isto é, pequena para ser descartada ou um retalho seja gerado, ou o padrão inicial tenha sido anulado. Neste último caso, o padrão de corte é definido pelo problema da mochila.

- Procedimento $Guloso_A$

O procedimento $Guloso_A$ consiste em resolver um problema da mochila para cada objeto disponível no estoque para obter um padrão de corte e, logo após, a sobra é analisada. Se a sobra for aceitável (pequena perda ou retalho), então o padrão é aceito, senão um item do padrão de corte (o maior) é retirado e a sobra é novamente analisada. Se a sobra gerada ainda não for aceitável, então outro item (segundo maior) é retirado do padrão de corte. Este processo é repetido até que sobra seja aceitável ou o padrão anulado. Se o padrão for anulado, escolhe-se entre os padrões originais, aquele que apresentar a menor perda (esta é uma situação atípica que pode ocorrer, por exemplo, quando o estoque é formado apenas por retalhos).

Heurísticas residuais também foram alteradas para resolver o problema de corte com sobras aproveitáveis. Estas heurísticas consistem em obter uma solução contínua a partir da relaxação do problema linear proposto por Gilmore e Gomory em 1963:

$$\text{minimizar } \sum_{k=1}^K \sum_{j=1}^{N_k} c_{jk} x_{jk} \quad (2.98)$$

sujeito a:

$$\sum_{k=1}^K \sum_{j=1}^{N_k} \alpha_{ijk} x_{jk} = d_i, \quad i = 1, \dots, m \quad (2.99)$$

$$\sum_{k=1}^{N_k} x_{jk} \leq e_k, \quad k = 1, \dots, K \quad (2.100)$$

$$x_{jk} \geq 0 \text{ e inteiro, } j = 1 \dots, N_k, \quad k = 1 \dots, K. \quad (2.101)$$

e utilizando alguma técnica de arredondamento obter uma solução inteira. Como esta estratégia pode deixar de atender algumas demandas devido ao arredondamento das soluções, o problema residual resultante é resolvido a partir de procedimentos heurísticos. Para o problema de aproveitamento de sobras, os autores alteraram as heurísticas Residuais FFD e Gulosa.

- Heurísticas Residuais FFD_A e Gulosa_A

As heurísticas residuais FFD_A e Gulosa_A consistem em obter uma solução inteira aproximada para (2.98)-(2.101) determinada por um truncamento trivial dado pelo inteiro inferior ao fracionário obtido. Com esta técnica, ao final do processo de corte, itens podem ter suas demandas não atendidas e, desta forma, os procedimentos heurísticos FFD_A e Gulosa_A são utilizados para resolver este problema residual.

Outros procedimentos heurísticos residuais alterados para resolver o aproveitamento de sobras foram propostos por Poldi e Arenales (2009). As Heurísticas Residuais de Arredondamento Guloso (RAG - versões 1, 2 e 3) originaram as heurísticas RAG_A - versões 1, 2 e 3 as quais são descritas a seguir.

- Heurísticas Residuais de Arredondamento (RAG_A - versões 1, 2 e 3)

Inicialmente uma solução inteira aproximada é obtida utilizando uma das versões da heurística RAG, que não necessita do problema residual. Em seguida a sobra em cada padrão é analisada. Se a sobra estiver em limites aceitáveis (calculados previamente), o padrão de corte é aceito e armazenado; caso contrário é rejeitado e em seguida desfeito (demanda e estoque são atualizados). Depois de analisados todos os padrões de corte

gerados, aplica-se a heurística FFD_A na demanda residual formada pelos padrões de corte rejeitados. Para este procedimento heurístico, o limiar para a perda aceitável é obtido pela perda percentual da solução inteira aproximada da heurística RAG (nas heurísticas construtivas, esse limitante é fornecido pelo usuário, para objetos padronizados e não padronizados).

Para verificar o desempenho dos procedimentos heurísticos, vários testes computacionais foram realizados, os quais compreendem exemplos práticos, da literatura e exemplos gerados aleatoriamente. Em todos os testes realizados, os procedimentos heurísticos que consideram o aproveitamento de sobras obtiveram um desempenho superior que suas versões clássicas sem o aproveitamento e o procedimento heurístico COLA (Gradisar *et al.*, 1997). A análise e classificação de todos os procedimentos heurísticos foram realizadas considerando critérios e definições estabelecidos a priori.

Todos os procedimentos heurísticos desenvolvidos resolvem problemas com altas dimensões em tempos computacionais muito baixo.

No decorrer desta tese, serão apresentados detalhes do problema estudado por Cherri *et al.* (2009), dos procedimentos heurísticos desenvolvidos para resolver o problema e dos testes computacionais realizados.

- **Koch *et al.* (2008) - Linear programming for a cutting problem in the wood processing industry - a case study**

O trabalho apresentado por Koch *et al.* envolve um estudo de caso na indústria madeireira no qual um modelo matemático foi desenvolvido para satisfazer os objetivos específicos da indústria que tem o processo de corte interligado ao transporte e manipulação do material. Além disso, após o processo de corte, as sobras geradas nos padrões de corte podem retornar ao estoque desde que o seu tamanho seja aceitável.

Embora a indústria atenda grandes pedidos de clientes, os autores consideram apenas os casos em que os pedidos são pequenos e podem ser atendidos diretamente com o material disponível no estoque que é composto por objetos padronizados e uma grande variedade de peças residuais (retalhos) gerados por cortes anteriores. Os objetos são estocados em compartimentos que são transportados até a área de corte. Cada compar-

timento possui objetos com qualidades e tamanhos diferentes. Após o processo de corte, os compartimentos com objetos não utilizados, retornam ao depósito.

Em geral, os itens a serem produzidos podem ser agrupados considerando as características estabelecidas pelos clientes. A demanda de todos os itens deve ser atendida e, além disso, custos de material (perda), transporte e manipulação devem ser minimizados. A quantidade de retalhos gerados durante o processo de corte também devem ser minimizadas, pois custos de transporte e estoque estão associados a estes retalhos.

Para evitar que grandes quantidades de perda sejam geradas, padrões de corte podem ser substituídos de modo a gerar retalhos. Assim, uma maior quantidade de objetos padronizados seria utilizada e os custos com a manipulação de objetos aumentaria. Além disso, vários custos seriam adicionados com a criação de retalho. Desta forma, os autores desenvolveram um aplicativo de suporte à decisão que controla a entrada de objetos padronizados e a criação de novos retalhos.

Para resolver este problema, a abordagem utilizada é baseada em programação linear, que possibilita ao tomador de decisão especificar o comprimento máximo aceitável para a perda e quais são os limites (inferior e superior) para o comprimento de um retalho. Ao final do processo de corte, os planos gerados com perdas e custos são apresentados ao tomador de decisão para que ele escolha qual solução será implementada.

Além da notação apresentada na seção 2.1.1, a modelagem matemática desenvolvida considera:

Conjunto de índices

- H : conjunto de índices dos números dos compartimentos;
- R : conjunto de índices dos intervalos que define os retalhos.

Constantes

- c^{hand} : custo de manipulação por compartimento;
- c^{waste} : custo da perda por unidade de comprimento;
- c^{resid} : custo dos retalhos por unidade de comprimento;

- c_{jk}^{trim} : custo da perda gerada pelo padrão de corte j para o objeto tipo k ;

$$c_{jk}^{trim} = \begin{cases} c^{waste}(L_k - \sum_{i=1}^m \ell_i \alpha_{ijk}), & \text{se a sobra no padrão } j \text{ é considerada perda} \\ c^{resid}(L_k - \sum_{i=1}^m \ell_i \alpha_{ijk}), & \text{se a sobra no padrão } j \text{ é considerada retalho} \\ 0, & \text{caso contrário.} \end{cases}$$

- e_{kh} : número de objetos tipo k disponível no compartimento h .

Variáveis

- y_h : indica a utilização do compartimento h ;

$$y_k = \begin{cases} 1, & \text{se pelo menos um objeto grande é utilizado no compartimento } k \\ 0, & \text{caso contrário.} \end{cases}$$

Parâmetros

- $WLMAX$: comprimento máximo para uma sobra ser aceita como perda;
- $[RLMIN_r, RLMAX_r]$: intervalo no qual uma sobra é definida como retalho.

Considerando estes parâmetros, as seguintes condições foram impostas:

$$WLMAX < \min(RLMN_r), \quad r \in R$$

$$RLMIN_r < RLMAX_r, \quad r \in R$$

Para que um padrão de corte $j, j = 1, \dots, N_k$, fosse considerado factível, uma das seguintes condições deveriam ser verificadas:

- uma sobra é ‘suficientemente pequena’ para ser aceita como perda:

$$0 \leq s_{jk} = L_k - \sum_{i=1}^m \ell_i \alpha_{ijk} \leq WLMAX,$$

- uma sobra tem comprimento para ser aceita como retalho:

$$RLMIN_r \leq s_{jk} = L_k - \sum_{i=1}^m \ell_i \alpha_{ijk} \leq RLMAX_r, r \in R.$$

Como as colunas são geradas a priori, os custos são conhecidos antes do problema ser resolvido. Assim, o problema foi modelado como o seguinte problema de otimização:

$$\text{minimizar } \sum_{k=1}^K \sum_{j=1}^{N_k} c_{jk}^{trim} x_{jk} + \sum_{h=1}^H c^{hand} y_h \quad (2.102)$$

sujeito a:

$$\sum_{k=1}^K \sum_{j=1}^{N_k} \alpha_{ijk} x_{jk} = d_i, \quad i = 1, \dots, m \quad (2.103)$$

$$\sum_{j=1}^{N_k} x_{jk} \leq \sum_{h=1}^H e_{kh} y_h, \quad k = 1, \dots, K \quad (2.104)$$

$$\sum_{j=1}^{N_k} x_{jk} \leq e_k, \quad k = 1, \dots, K \quad (2.105)$$

$$x_{jk} \geq 0 \text{ e inteiro}, \quad j = 1, \dots, N_k \quad k = 1, \dots, K \quad (2.106)$$

$$y_h \in \{0, 1\}, \quad h = 1, \dots, H. \quad (2.107)$$

A função objetivo (2.102) minimiza os custos relevantes do problema e as demais restrições são referentes a demanda (2.103) e estoque dos objetos (2.104)-(2.105). A implementação desenvolvida considera os dados da indústria, assim como parâmetros que são fornecidos pelo tomador de decisão que avalia as soluções que foram computadas para diferentes valores estabelecidos e seleciona uma para execução.

Segundo os autores, as soluções obtidas foram bem aceitáveis pela empresa.

2.1.4 Metaheurística

Metaheurística pode ser vista como uma estrutura de algoritmo geral que pode ser aplicado a diferentes problemas de otimização com relativamente poucas modificações a serem feitas para adaptá-los a um problema específico. Foi encontrado apenas um artigo na literatura que trata o problema de corte de estoque com sobras aproveitáveis e utiliza metaheurística como método de solução. Este artigo é comentado a seguir.

- **Kos e Duhovnik (2002) - Cutting optimization with variable-sized stock and inventory status data**

Neste trabalho, Kos e Duhovnik apresentam um algoritmo genético híbrido para resolver o problema de corte de estoque que surge particularmente na produção de estruturas de aço. O estudo desenvolvido tem como objetivo a minimização das perdas sendo que o estoque é formado por objetos de diferentes comprimentos. Assim, os autores consideraram a possibilidade de sobras retornarem ao estoque, desde que fossem superiores a um determinado limitante fornecido pelo usuário, baseado no tipo e planejamento da produção.

Considerando o fato de que os objetos a serem cortados são de tamanhos diferentes, os autores utilizaram a formulação *Variable-Sized Bin Packing Problem* (VBPP) para resolver o problema. Essa formulação é uma extensão da formulação clássica de *Bin Packing Problem* (BPP), que considera objetos em estoque com apenas um comprimento.

O método de solução proposto cria uma codificação de cromossomos e operadores genéticos satisfatórios para o VBPP. A codificação do grupo de *bins* foi estendida para controlar diferentes capacidades, que resultou em novos operadores do Algoritmo Genético VBPP, uma nova função de avaliação, procedimento de adaptação e outras partes da estrutura do algoritmo genético.

Um cromossomo pode ser representado por uma série de *bins*, cada qual com as propriedades de *capacidade* (define a soma máxima do tamanho dos itens) e *itens* (lista indexada com referências dos tamanhos dos itens a serem alocados nos *bins*).

O algoritmo genético desenvolvido para o VBPP tem a seguinte estrutura geral:

Passo 1: Uma população inicial é gerada como um conjunto de cromossomos válidos;

Passo 2: Avaliação da função de aptidão de cada indivíduo;

Passo 3: Seleção (duplicação) de indivíduos para gerar uma população intermediária;

Passo 4: Cruzamento de dois indivíduos para gerar filhos;

Passo 5: Mutação em alguns indivíduos (a fim de introduzir diversidade na próxima população);

Passo 6: Verificação do critério de parada, que é a função de aptidão definida no Passo

2. Se o melhor indivíduo satisfaz o critério de parada, então o Algoritmo Genético pára, senão, continua no Passo 3.

Em geral, Algoritmos Genéticos usam dois tipos de operadores, o cruzamento e a mutação.

- *Cruzamento*: O cruzamento é uma operação realizada para definir novos indivíduos que tenham as características genéticas de seus pais. O cruzamento ocorre em um par de cromossomos (chamados de pais) e gera um novo cromossomo (chamado filho). A estratégia utilizada foi a de cruzamento em dois pontos.
- *Mutação*: A mutação é um operador utilizado para perturbar a convergência de uma população, para obter uma busca mais ampla. Alguns cromossomos são selecionados aleatoriamente e apenas um de seus gens é modificado. A taxa de mutação é geralmente baixa, para evitar discrepâncias.

Como os autores consideram o aproveitamento de sobras, para manter o estoque com quantidades razoáveis de retalhos estes objetos têm prioridade de uso em relação aos demais objetos do estoque (objetos padronizados).

A otimização dos retalhos pode ser vista como um pré-processamento na resolução do VBPP que tenta utilizar o máximo possível dos retalhos disponíveis. O algoritmo de otimização dos retalhos consiste basicamente nos seguintes passos:

Passo 1: Encontrar o menor item a ser cortado/empacotado. Selecionar os objetos que tenham capacidade de acomodar esse item. Desconsiderar os itens que sejam maiores que o maior retalho disponível no estoque, esses itens serão tratados pelo algoritmo VBPP;

Passo 2: Para o conjunto reduzido de itens, aplique o algoritmo para o MSSP (*Multiple Subset Sum Problem*);

Passo 3: Os itens que não foram alocados nos Passos 1 e 2, devem ser alocados agora aplicando-se o algoritmo VBPP.

O *Multiple Subset Sum Problem* (MSSP) é um caso especial do *Multiple Knapsack Problem* (MKP), em que o número de *bins* disponíveis é fixo e pode não ser suficiente

para alocar todos os itens. Assim, apenas um subconjunto dos itens é alocado e os itens que ficaram de fora serão alocados no Passo 3, usando-se o algoritmo VBPP.

Para realizar testes computacionais, os autores utilizaram conjuntos de dados de dois artigos da literatura, porém ambos fornecem dados para o problema de corte de estoque clássico. Assim, tais dados foram adaptados para o problema de corte de estoque com sobras aproveitáveis.

Segundo os autores, o método proposto apresentou boas soluções para o problema.

Capítulo 3

O problema de corte de estoque com sobras de material aproveitáveis

Neste capítulo definimos o problema de corte de estoque com sobras de material aproveitáveis nos casos unidimensional e bidimensional. No caso unidimensional apresentamos duas abordagens para o problema. Na primeira abordagem, definimos o problema de corte com sobras aproveitáveis, em que o principal objetivo é a minimização das perdas e os retalhos disponíveis no estoque têm um simples incentivo de uso. Na segunda abordagem, além da minimização das perdas, os retalhos disponíveis em estoque têm prioridade de corte em relação aos demais objetos. Além disso, um tamanho máximo para uma perda aceitável e um tamanho mínimo para um retalho são usados para aceitar ou não (isto é, rejeitar ou reformular parcialmente) padrões de corte gerados por métodos clássicos.

3.1 Problema de corte unidimensional com sobras aproveitáveis - primeira abordagem

Durante o processo de corte de peças, sobras inevitáveis ocorrem e, eventualmente, são descartadas. Porém, algumas indústrias apresentam a possibilidade de utilizar estas sobras para cortes futuros, desde que tenham tamanhos significativos.

Muitos dos métodos de solução para os problemas de corte buscam minimizar sobras, (objetivos alternativos podem ser definidos, mas sobras baixas devem ser perse-

guidas) sendo que, nesses métodos, considera-se como sobra todo pedaço cortado que não seja um item demandado. Embora sobras baixas seja ainda um objetivo perseguido, a possibilidade de aproveitá-las para atender demandas futuras introduz uma nova condição na avaliação de uma solução. Planejar padrões de corte que concentrem as sobras em poucos padrões parece ser uma boa alternativa a ser perseguida, pois aumenta as chances delas serem suficientemente grandes para voltarem ao estoque e serem melhor utilizadas no futuro.

Veja por exemplo, as soluções 1 e 2 (Figura 3.1 - c e d). Ambas minimizam o comprimento total dos objetos cortados (objetivo freqüentemente utilizado). Entretanto, a Solução 2 concentra as sobras em um único objeto, permitindo que esta sobra seja utilizada no futuro. Porém, nem sempre um simples rearranjo dos itens entre os objetos cortados permite a obtenção de um retalho (isto é, sobra suficientemente grande) e uma nova questão se impõe: compensa cortar parcialmente alguns objetos a mais de modo a reduzir as perdas para os dados da demanda atual e gerar retalhos para demandas futuras?

O problema de corte de estoque unidimensional com sobras de material aproveitáveis é definido como:

*Um conjunto de peças (itens) deve ser produzido a partir do corte de unidades grandes (objetos), os quais podem ser de tamanhos padronizados (objetos que são comprados de fornecedores) ou não padronizados (objetos que são retalhos de cortes anteriores). São dados os tamanhos e as quantidades dos itens demandados e dos objetos disponíveis no estoque. As demandas devem ser atendidas, cortando-se os objetos disponíveis, de modo que as sobras sejam ‘pequenas’ (chamadas de **perdas**) ou ‘suficientemente grandes’ (chamadas de **retalhos**) para retornarem ao estoque, porém em número reduzido.*

Esta definição, em alto nível, procura captar os principais elementos do problema de corte com sobras aproveitáveis, porém necessita de detalhes que são completados a seguir, sendo gradativamente quantificados.

Um comprimento ‘suficientemente grande’ ou, de outra forma, um comprimento mínimo aceitável para um retalho é uma decisão que pode ser tomada pelo usuário.

Algumas possíveis escolhas incluem: o comprimento do menor item demandado, a média dos comprimentos dos itens demandados, o comprimento do maior item, ou qualquer valor arbitrário definido pelo usuário. Gradisar *et al.* (1997), (1999a, b), Gradisar e Trkman (2005) e Abuabara e Morabito (2008, 2009) consideram como retalho qualquer sobra de comprimento maior ou igual ao comprimento do menor item demandado. A escolha do menor item, como sugerido na literatura, pode não ser interessante, pois uma carteira de pedidos pode ter um item atípico pequeno e retalhos de pouco uso seriam estocados. Também uma carteira particular pode ter somente itens de comprimentos grandes, de modo que perdas menores que o menor item demandado seriam aceitáveis como retalhos futuros.

Diferentemente dos problemas clássicos de corte, para os quais funções objetivos são bem definidas, no problema de corte com sobras de material aproveitáveis objetivamos sobras ‘pequenas’ (como nos problemas clássicos), e/ou ‘suficientemente grande’, porém em um número reduzido, pois uma quantidade excessiva de retalhos pode tornar a solução operacionalmente inviável. Duas soluções com a mesma sobra são, agora, diferenciadas. Para uma melhor compreensão do problema de corte de estoque com sobras de material aproveitáveis, considere o seguinte exemplo, no qual estabelecemos que toda sobra de tamanho superior ou igual a 4 metros é considerada retalho.

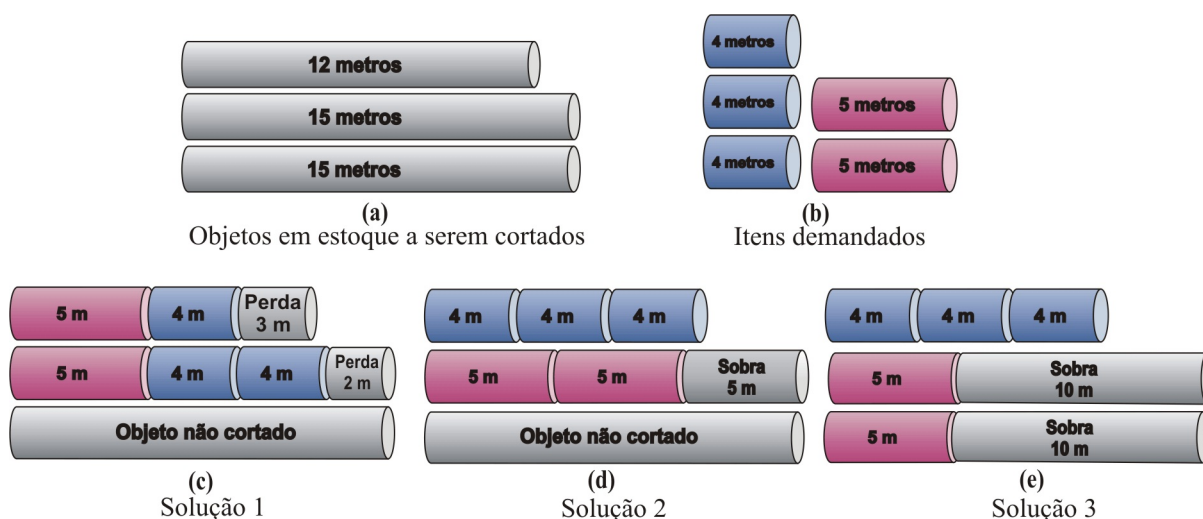


Figura 3.1: Dados de um problema de corte unidimensional e soluções alternativas.

Para o problema de corte de estoque com sobras de material aproveitáveis, a Solução 2 (Fig. 3.1 - **d**) é melhor que a Solução 1 (Fig. 3.1 - **c**), pois concentra as sobras em um único objeto e, como é superior a 4 metros, é um retalho que poderá ser utilizado para atender demandas futuras (a Solução 1 tem perda de 5 metros enquanto que a Solução 2 tem perda zero e um retalho de 5 metros). Para o problema de corte com sobras aproveitáveis podemos dizer que a Solução 1 é uma solução *indesejável*, enquanto a Solução 2 é *ideal*. A Solução 3 também pode ser considerada *indesejável*, pois embora não gere perdas, apresenta um número maior de retalhos, o qual também deve ser reduzido.

Devido à dificuldade em definir uma função objetivo que diferencie as soluções do problema de corte com sobras de material aproveitáveis, qualificamos as soluções conforme a definição 3.1:

Definição 3.1 *Para o problema de corte de estoque com sobras de material aproveitáveis, as soluções são definidas como:*

- **Solução ideal:** *quando poucos objetos cortados apresentarem perda pequena e nenhum dos objetos cortados apresentar perda não tão pequena (não aproveitáveis). Retalhos devem estar concentrados em muito poucos objetos cortados;*
- **Solução aceitável:** *quando poucos objetos cortados apresentarem perdas não tão pequenas e poucos retalhos;*
- **Solução indesejável:** *quando vários objetos cortados apresentarem perdas não tão pequenas ou vários retalhos.*

Nesta definição (que depende ainda da quantificação dos termos *poucos*, *muito poucos* ou *vários objetos*, *perda pequena* ou *não tão pequena* e *retalho*), procuramos captar características genéricas para uma boa solução do problema de corte de estoque com sobras aproveitáveis. Uma perda é considerada ‘*não tão pequena*’ desde que ela seja maior que uma *perda pequena* porém não grande o suficiente para ser um *retalho*. A Figura 3.2 ilustra esta classificação:

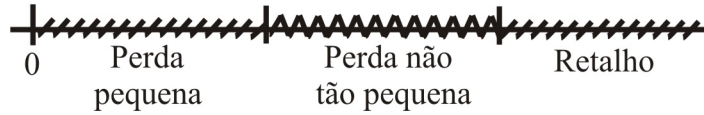


Figura 3.2: Classificação das sobras.

A quantificação de *perda pequena*, *não tão pequena* ou *retalho* pode ser feita pelo usuário (decisor) de acordo com sua experiência. O usuário também pode utilizar os seguintes parâmetros:

- θ : fração que define o tamanho máximo para que uma *sobra* seja considerada *perda pequena* em um objeto padronizado, ou seja, θL_k é o tamanho máximo para que uma sobra seja considerada pequena em um objeto de comprimento $L_k, k = 1, \dots, \bar{k}$ em que \bar{k} é a quantidade de tipos de objetos padronizados disponíveis no estoque;
- β : fração que define o tamanho máximo para que uma *sobra* seja considerada *perda pequena* em um objeto não padronizado, ou seja, βL_k é o tamanho máximo para que uma sobra seja considerada pequena em um objeto não padronizado de comprimento $L_k, k = \bar{k} + 1, \dots, K$ (os objetos do tipo $\bar{k} + 1, \dots, K$ são retalhos e K é a quantidade total de tipos de objetos padronizados e não padronizados disponíveis no estoque);
- δ : Tamanho mínimo para que uma *sobra* seja considerada *retalho* (p. ex., δ pode ser o comprimento do menor item demandado ou a média aritmética dos comprimentos dos itens demandados). Qualquer sobra em um padrão de corte maior ou igual a δ é considerado retalho, independentemente do tipo de objeto.

Observe que, com os parâmetros θ e β , tornamos a perda dependente do objeto a ser cortado. O parâmetro adicional β nos permite definir perdas percentualmente maiores para os objetos não padronizados. Isto pode fazer com que estes objetos tenham mais chances de uso em relação aos objetos padronizados.

Os termos *muito poucos*, *poucos* ou *vários* na Definição 3.1 são também definidos pelo usuário (decisor). O usuário pode definir os termos por experiência ou utilizar, por exemplo, os parâmetros: ε_1 e ε_2 com $0 < \varepsilon_1 < \varepsilon_2 < 1$ e definir:

- *Muito poucos objetos cortados*: até $\lceil \varepsilon_1 \eta \rceil$ dos objetos cortados;

- *Poucos objetos cortados*: até $\lceil \varepsilon_2 \eta \rceil$ dos objetos cortados;
- *Vários objetos cortados*: acima de $\lceil \varepsilon_2 \eta \rceil$ dos objetos cortados;

em que η é o número total de objetos cortados numa solução.

Por simplicidade, utilizaremos em todo texto o termo *sobra aceitável* quando esta for uma *perda pequena* ou um *retalho*.

3.2 Problema de corte unidimensional com sobras aproveitáveis e prioridade no uso dos retalhos - segunda abordagem

O problema de corte de estoque com sobras de material aproveitáveis apresentado na Seção 3.1 tem como objetivo a minimização das sobras e os retalhos disponíveis em estoque têm como incentivo de uso o parâmetro β , o qual permite que perdas maiores sejam geradas nestes objetos. Entretanto, este incentivo de uso pode não ser suficiente para manter o estoque de retalhos reduzido.

Em muitos processos industriais, dependendo do tipo de material utilizado (bobinas de aço, por exemplo), retalhos estocados podem tornar-se sucatas, caso a empresa não utilize estes objetos após um determinado período de tempo.

Considerando esta situação, focamos o problema de corte de estoque com sobras de material aproveitáveis em que, adicional à minimização das sobras, os retalhos disponíveis em estoque têm prioridade de uso em relação aos objetos padronizados durante todo o processo de corte que ocorre em um determinado período de tempo, ou seja, os retalhos gerados em períodos anteriores do processo de corte devem ser utilizados com uma certa prioridade.

A descrição do problema de corte de estoque com sobras de material aproveitáveis e prioridade no corte de objetos não padronizados é semelhante a apresentada na Seção 3.1, com algumas alterações apresentadas a seguir:

Um conjunto de itens, os quais têm comprimentos e demandas conhecidas para o período atual, deve ser produzido a partir do corte de objetos disponíveis em

estoque (inicialmente somente objetos padronizados). Os objetos não utilizados no período atual ficam disponíveis para o próximo período juntamente com os retalhos gerados. As demandas dos itens do período atual devem ser atendidas, cortando-se os objetos disponíveis, de modo a minimizar as perdas e admitindo-se a geração de retalhos (comprimentos de objetos não utilizados, suficientemente grandes). Os objetos não padronizados (retalhos) disponíveis em estoque no período atual devem ser usados prioritariamente.

Nesta descrição do problema, observe que não existe apenas um único objetivo a ser satisfeito, uma vez que desejamos minimizar as sobras (como nos problemas clássicos) e reduzir a quantidade de objetos não padronizados disponíveis no estoque.

Neste problema, a quantificação das perdas e do retalho também pode ser feita pelo usuário (decisor) de acordo com sua experiência. Por simplicidade, para quantificar os termos *perda* e *retalho*, utilizamos os parâmetros θ e δ definidos na Seção 3.1 e o parâmetro β é definido como:

- β : tamanho máximo para que uma *sobra* seja considerada *perda* em um objeto não padronizado (retalho), ou seja, objeto de comprimento L_k , $k = \bar{k} + 1, \dots, K$ (os objetos do tipo $k = \bar{k} + 1, \dots, K$ são retalhos), em que K é a quantidade total de tipos de objetos padronizados e não padronizados disponíveis no estoque.

Note que para os objetos padronizados mantemos o critério anterior (θL_k , $k = 1, \dots, \bar{k}$) para definir uma *perda* (aceitável), enquanto que para os objetos não padronizados, definimos um valor fixo β . Como os retalhos são gerados a partir do corte de objetos padronizados com perdas nulas (as perdas são somadas ao retalho), estabelecemos um valor β que se aproxima de θL_k , $k = 1, \dots, \bar{k}$, pois, com o parâmetro β definido na Seção 3.1, os objetos não padronizados menores eram somente cortados com uma perda muito pequena, a qual representava uma porcentagem muito baixa quando comparada com a porcentagem θ , estimada para o objeto padronizado do qual o retalho foi gerado.

Para uma melhor compreensão desta nova abordagem para o problema de corte de estoque com sobras de material aproveitáveis, considere o seguinte exemplo, no qual temos 1 período de tempo e estabelecemos que toda sobra de tamanho superior ou igual

a 4 metros (média do comprimento dos itens demandados) é considerada retalho. Os objetos de comprimento $L_k, k = 2, 3$ e 4, são considerados objetos não padronizados (retalhos gerados em períodos anteriores).

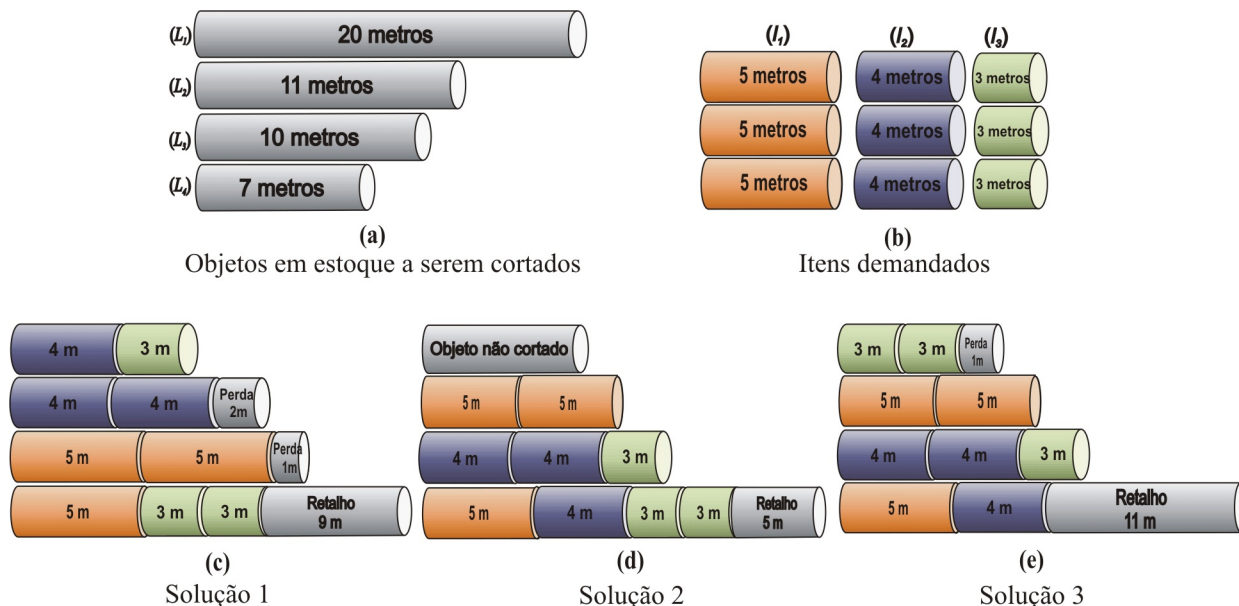


Figura 3.3: Dados de um problema de corte unidimensional e soluções alternativas.

Neste exemplo, observamos que a Solução 1 (Fig. 3.3 - c) e a Solução 3 (Fig. 3.3 - e) cortam todos os retalhos do estoque e geram um novo que deve retornar ao estoque. Desta forma, como a perda total na Solução 3 é menor, podemos dizer que esta solução é preferida em relação a Solução 1. Por outro lado a Solução 2 (Fig. 3.3 - d) tem perda nula, entretanto, não prioriza o corte dos retalhos em estoque, gerando para o próximo período 2 retalhos (um retalho novo e outro de períodos anteriores). Considerando nossa abordagem e as três soluções apresentadas, temos que a Solução 3 é a mais adequada.

3.3 Problema de corte bidimensional com sobras aproveitáveis

Os métodos existentes na literatura para resolver o problema de corte bidimensional têm como principal objetivo minimizar a sobra resultante nos objetos cortados. Entretanto, assim como nos problemas de corte unidimensionais, os padrões de corte

podem ser planejados de modo a admitir que sobras grandes o suficiente sejam utilizadas no futuro.

De modo similar ao caso unidimensional (segunda abordagem), descrevemos o problema de corte de estoque bidimensional com sobras de material aproveitáveis como:

Um conjunto de peças retangulares (itens), as quais possuem dimensões e demandas conhecidas para o período atual, deve ser produzido a partir do corte de placas retangulares (objetos) disponíveis em estoque (inicialmente somente placas padronizadas). As placas não utilizadas no período atual ficam disponíveis para o próximo período juntamente com os retalhos gerados. As demandas dos itens do período atual devem ser atendidas, cortando-se as placas disponíveis, de modo a minimizar as perdas e admitindo-se a geração de retalhos (sobras com dimensões suficientemente grandes). As placas não padronizadas (retalhos) disponíveis em estoque no período atual devem ser usadas prioritariamente.

Para classificar uma sobra como ‘perda’ ou ‘retalho’, dados podem ser fornecidos pelo usuário com base em suas experiências. Neste trabalho, definimos os termos *perda* ou *retalho* de acordo com alguns critérios, os quais são tratados separadamente para objetos padronizados e não padronizados. Esta separação foi realizada pelo fato da sobra ser analisada por parâmetros e critérios diferentes em cada classe de objetos (padronizados e não padronizados). Desta forma, temos:

• Retalhos e perdas em objetos padronizados

Seja $(\bar{\ell}, \bar{w})$ uma sobra na placa k ($k = 1, \dots, \bar{k}$) e $A_s = (\bar{\ell} \times \bar{w})$ a área correspondente.

Retalho: Para definir um retalho, utilizamos apenas um critério.

Se

$$\bar{\ell} \geq \phi L_k \quad \text{e} \quad \bar{w} \geq \phi W_k \tag{3.1}$$

então a sobra $(\bar{\ell}, \bar{w})$ é um retalho e ϕ , é tal que $0 < \phi < 1$.

Ao invés das dimensões mínimas para definir o retalho, dadas por ϕL_k e ϕW_k , podemos utilizar as dimensões do item de menor área, ou qualquer valor fornecido pelo usuário.

Perda: Utilizamos dois critérios para definir uma *perda aceitável* (isto é, perda considerada pequena para ser descartada), a primeira é baseada na área e a segunda em suas dimensões.

- *Primeiro critério:* Se

$$A_s < \xi L_k W_k \quad (3.2)$$

então a sobra $(\bar{\ell}, \bar{w})$ é uma perda aceitável. Em 3.2, ξ é tal que $0 < \xi < 1$.

Se $A_s \geq \xi L_k W_k$, ou seja, a sobra tem uma área relativamente grande, verificamos ainda suas dimensões.

- *Segundo critério:* Se

$$\bar{\ell} \leq \rho L_k \quad \text{ou} \quad \bar{w} \leq \rho W_k \quad (3.3)$$

então a sobra $(\bar{\ell}, \bar{w})$ é uma perda aceitável e, $0 < \rho < 1$.

Se uma sobra não satisfaz os critérios (3.1), que define um retalho ou (3.2) - (3.3) que definem uma perda aceitável, então a sobra é considerada *perda indesejável*.

• Retalhos e perdas em objetos não padronizados

Os objetos não padronizados $k, k = \bar{k} + 1, \dots, K$, em estoque (os objetos do tipo $\bar{k} + 1, \dots, K$ são retalhos e K é a quantidade total de tipos de objetos padronizados e não padronizados disponíveis no estoque) são tratados de maneiras distintas e, portanto, são divididos em dois grupos:

- *objetos não padronizados grandes* $k, k = \bar{k} + 1, \dots, P$: objetos com área maior ou igual a 50% da área do menor objeto padronizado em estoque;
- *objetos não padronizados pequenos* $k, k = P + 1, \dots, K$: objetos com área inferior a 50% da área do menor objeto padronizado em estoque.

Para os objetos não padronizados $k, k = \bar{k} + 1, \dots, P$, ou seja, objetos não padronizados grandes, utilizamos os mesmos critérios (3.1),(3.2) e (3.3) para a analisar a sobra e classificá-la como *retalho*, *perda aceitável* ou *perda indesejável*.

Para os objetos não padronizados pequenos ($k = P + 1, \dots, K$), utilizamos apenas um critério para definir perda ou retalho.

Seja $(\bar{\ell}_s, \bar{w}_s)$ dimensões estabelecidas a priori e, $(\bar{\ell}, \bar{w})$ uma sobra na placa k ($k = P + 1, \dots, K$). Se

$$\bar{\ell} \geq \bar{\ell}_s \text{ e } \bar{w} \geq \bar{w}_s \quad (3.4)$$

então a sobra $(\bar{\ell}, \bar{w})$ é um *retalho*, caso contrário, é uma *perda aceitável*.

As dimensões que classificam uma sobra como retalho é sempre a mesma para todos os objetos não padronizados pequenos ($k = P + 1, \dots, K$) do estoque. Observe que para estes objetos, uma sobra não é classificada como *perda indesejável*.

Para uma melhor compreensão do problema de corte bidimensional com sobras de material aproveitáveis, considere o seguinte exemplo, no qual estabelecemos que toda sobra com dimensões superiores a (15×10) é um retalho que pode ser utilizado no atendimento de demandas futuras e as dimensões dos itens a serem cortados são:

- $(\ell_1 \times w_1) = (15 \times 25)$;
- $(\ell_2 \times w_2) = (21 \times 14)$;
- $(\ell_3 \times w_3) = (12 \times 13)$;
- $(\ell_4 \times w_4) = (24 \times 21)$.

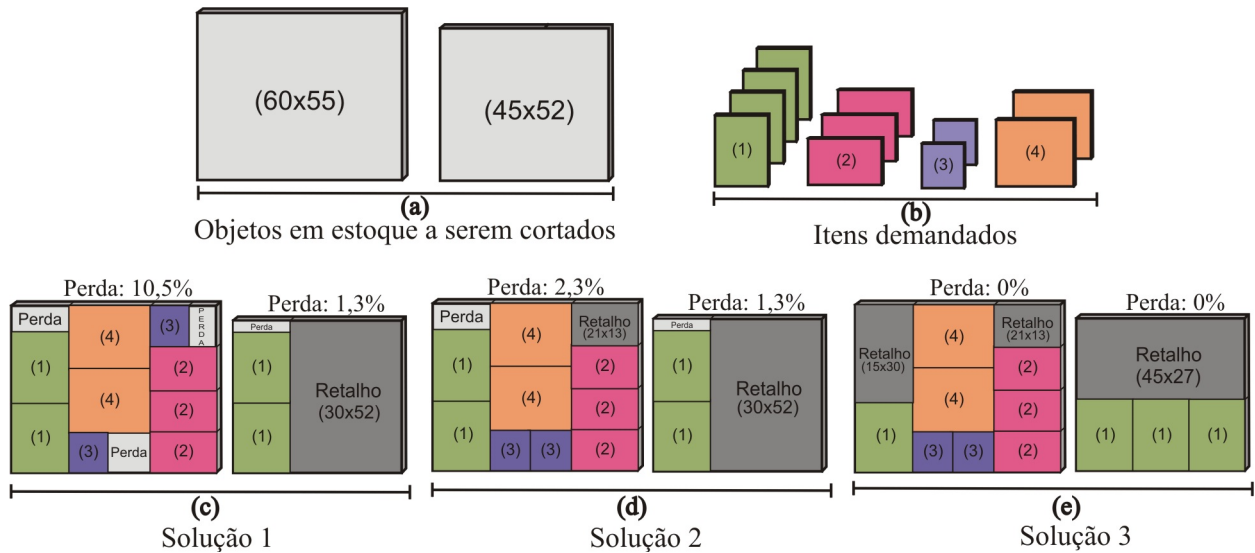


Figura 3.4: Dados de um problema de corte bidimensional e soluções alternativas.

Neste exemplo, observamos que os três possíveis padrões de corte são equivalentes se avaliarmos a função objetivo *sobra total* (soma das áreas das sobras nos objetos cortados). Entretanto, a Solução 2 (Fig. 3.4 - d) deve ser preferível à Solução 1 (Fig. 3.4 - c), uma vez que se trata de um rearranjo de itens dentro de um padrão de corte. Com relação a Solução 3 (Fig. 3.4 - e), obtida por desfazer parcialmente a Solução 2, observamos que a perda foi reduzida, porém, o número de retalhos aumentou, ou seja, esta solução adia o desperdício com a demanda atual, na expectativa de que demandas futuras possam ser mais bem adequadas aos retalhos gerados, reduzindo a perda total num horizonte de planejamento, além do período atual.

Considerando os problemas descritos (Seções 3.1, 3.2 e 3.3), introduzimos alterações em alguns procedimentos heurísticos existentes na literatura que resolvem problemas clássicos de corte de estoque. Nosso objetivo com as alterações realizadas é gerar soluções para os problemas de corte (unidimensional e bidimensional) quando sobras podem retornar ao estoque. Os procedimentos heurísticos gerados para resolver os problemas de corte de estoque com sobras aproveitáveis são descritos nos Capítulos 4 e 5.

Capítulo 4

Métodos de solução para o problema de corte unidimensional com sobras aproveitáveis

Neste capítulo apresentamos os procedimentos heurísticos desenvolvidos para resolver o problema de corte de estoque unidimensional com sobras de material aproveitáveis. Tais procedimentos foram obtidos realizando modificações em procedimentos heurísticos clássicos bem conhecidos na literatura para o problema de corte de estoque. Para estes procedimentos, também fazemos uma breve revisão.

4.1 Heurísticas Construtivas

Uma heurística utilizada para encontrar a solução dos problemas de corte de estoque é a de repetição exaustiva Hinxman (1980).

Algoritmo - Heurística de repetição exaustiva

Passo 1: Construa um bom padrão de corte para cada objeto do tipo $k, k = 1, \dots, K$;

Passo 2: Escolha um dos padrões de corte gerados no Passo 1 (um critério de seleção pode ser, por exemplo, perda mínima. Este padrão selecionado está associado à um tipo de objeto);

Passo 3: Use o padrão de corte escolhido no Passo 2 o máximo possível, sem exceder a

demanda dos itens e a disponibilidade do objeto associado;

Passo 4: Atualize a demanda dos itens e o estoque dos objetos;

Passo 5: Se as demandas dos itens foram atendidas ou não houver mais disponibilidade de objetos então PARE. Senão, vá para o Passo 1.

O desempenho das heurísticas de repetição exaustiva dependem fundamentalmente do procedimento que gerar “bons” padrões de corte (Passo 1). Dois procedimentos bem conhecidos na literatura para gerar padrões de corte são: FFD (*First Fit Decreasing*) e Guloso.

4.1.1 Procedimento FFD

O procedimento FFD consiste em cortar primeiro o maior item tantas vezes quanto for possível, ou até que sua demanda já tenha sido atendida (os itens maiores são escolhidos em primeiro lugar, pois são, geralmente, mais difíceis de serem combinados). Assim, quando não for mais possível ou necessária a produção do maior item, o segundo maior item é considerado e, assim por diante, até o menor item. Quando nenhum novo item puder ser cortado, um padrão de corte é construído.

4.1.2 Procedimento Guloso

No procedimento guloso, os padrões de corte são gerados pela resolução sucessiva de Problemas da Mochila da forma:

$$\begin{aligned} z(b) = \text{maximizar} \quad & \ell_1 a_1 + \ell_2 a_2 + \dots + \ell_m a_m \\ \text{sujeito a:} \quad & \begin{cases} \ell_1 a_1 + \ell_2 a_2 + \dots + \ell_m a_m \leq b \\ 0 \leq a_i \leq r_i, \quad i = 1, \dots, m \text{ e inteiro} \end{cases} \end{aligned} \quad (4.1)$$

em que ℓ_i é o comprimento do item i , $i = 1, \dots, m$, e r_i é a demanda residual do item i , $i = 1, \dots, m$, atualizada no Passo 4 do algoritmo de repetição exaustiva. Inicialmente, $r_i = d_i$, $i = 1, \dots, m$, ou seja, r_i é a demanda inicial de itens tipo i .

Esses dois procedimentos clássicos têm filosofias distintas. No procedimento FFD, há uma preocupação excessiva de produzir o quanto antes os itens maiores, já que são difíceis de serem combinados, enquanto que no procedimento Guloso escolhe-se os me-

lhores padrões de corte possíveis para serem gerados primeiro, sem se preocupar com os padrões de corte futuros.

Para resolver o problema de corte com sobras aproveitáveis, realizamos modificações nestes dois procedimentos heurísticos. Com as alterações realizadas, desenvolvemos as heurísticas FFD_A e $Gulosa_A$. Estas heurísticas, que têm como objetivo a minimização das sobras, consideram o aproveitamento de sobras sem priorizar o corte de objetos não padronizados.

4.1.3 Heurística FFD_A

Para obter FFD_A , modificamos o procedimento FFD, de modo a evitar perdas não tão pequenas e, portanto, tentar obter soluções, pelo menos *aceitáveis*.

O procedimento FFD_A consiste em aplicar FFD para obter um padrão de corte e, logo após, a sobra é analisada. Se a sobra for aceitável (perda pequena ou retalho), o padrão é aceito, senão um item do padrão (o maior) é retirado. Assim, para o espaço gerado com a retirada do item é resolvido o problema da mochila (4.1), com capacidade b igual a sobra de material gerada no padrão de corte adicionada do tamanho do item retirado. Depois de resolvido o problema da mochila, a sobra gerada é analisada e se não for aceitável, outro item do padrão inicial (segundo maior) é retirado. Novamente para o espaço gerado é resolvido um novo problema da mochila (4.1). Caso tenhamos retirado um item de cada comprimento dentre todos que compõem o padrão, voltamos a retirar o primeiro maior. Este procedimento é repetido até que a sobra obtida seja aceitável, ou o padrão inicial tenha sido anulado. Neste último caso, o padrão de corte é definido pelo problema da mochila (4.1).

A seguir apresentamos os principais passos do algoritmo FFD_A . Neste algoritmo, denotamos por e_k a quantidade de objetos do tipo k , $k = 1, \dots, K$, disponíveis no estoque e d_i a demanda do item tipo i , $i = 1, \dots, m$.

Algoritmo FFD_A

$k = 1$

Passo 1: $\{Início\}$

Se $e_k = 0$ então $k = k + 1$, vá para o Passo 9;

Senão

aplique o procedimento FFD para um objeto do tipo k e obtenha o padrão FFD, cujo vetor associado é α_k^{FFD} .

Passo 2: *{Análise da sobra - padrão FFD}*

Se a sobra for aceitável (perda pequena ou retalho) então

Início

padrão α_k^{FFD} é aceito {temporariamente armazenado na lista B};

$k = k + 1$, vá para o Passo 9.

Fim

Passo 3: *{Item a ser retirado do padrão FFD}*

i = índice do item de maior comprimento em α_k^{FFD} .

Passo 4: *{Altera padrão com perda indesejável}*

Retire uma unidade do item i de α_k^{FFD} ;

ESPAÇO = sobra do padrão FFD + ℓ_i ;

Resolva o Problema da Mochila (4.1) com capacidade $b = \text{ESPAÇO}$, considerando os itens que já estão no padrão FFD e obtenha o padrão mochila, cujo vetor associado é denotado por $\alpha^{mochila}$.

Passo 5: *{Seleção do padrão de corte quando todos os itens são retirados do padrão FFD}*

Se $\alpha_k^{FFD} = 0$, então

Início

Enquanto a sobra não for aceitável faça:

Início

Seja p o índice do item de maior comprimento no padrão $\alpha^{mochila}$;

Retire uma unidade do item p de $\alpha^{mochila}$;

Fim

Se $\alpha^{mochila} \neq 0$ então

Início

padrão $\alpha_k^{mochila}$ é aceito {temporariamente armazenado na lista B};

Fim

Senão

Início

padrão $\alpha^{mochila}$ obtido no Passo 4 é aceito {temporariamente armazenado na lista C};

Fim

$k = k + 1$, vá para o Passo 9.

Fim

Passo 6: {*Novo padrão*}

Novo padrão (FFD + mochila): $\alpha_k^{FFD} + \alpha^{mochila}$.

Passo 7: {*Análise da sobra: padrão FFD + mochila*}

Se a sobra for aceitável então

Início

padrão $\alpha_k^{FFD} + \alpha^{mochila}$ é aceito {temporariamente armazenado na lista B};

$k = k + 1$, vá para o Passo 9.

Fim

Passo 8: {*Item a ser retirado do padrão FFD - atualização do índice*}

i = próximo item a ser retirado de α_k^{FFD} de acordo com o Procedimento **Retirada de Itens** (o padrão FFD + mochila é desconsiderado);

Vá para o Passo 4.

Passo 9: {*Determinação do melhor padrão de corte, frequência e atualização da demanda e do estoque*}

Se $k = K + 1$ então

Início

Se a lista $B = \emptyset$ então

Início

Escolha entre os padrões de corte da lista C aquele que apresentar menor perda e armazene;

Fim

Senão

Início

Escolha entre os padrões de corte da lista B aquele que apresentar menor perda ou se não existir padrão com perda, aquele que apresentar menor o retalho e armazene;

Fim

Determine a frequência do padrão escolhido (respeitando-se as restrições de demanda e estoque) e armazene;

Atualize $d_i, i = 1, \dots, m$ e $e_k, k = 1, \dots, K$;
 Se $d_i = 0, i = 1, \dots, m$ (solução factível encontrada) ou $e_k = 0, k = 1, \dots, K$ (demanda não atendida) então PARE;
 Senão $k = 1$ e esvazie as listas B e C;
 Fim
 Vá para o Passo 1.

Procedimento Retirada de Itens: Retira-se os itens maiores em ordem não-crescente de tamanho, um de cada tipo por vez. Caso tenha-se tirado um item de cada tipo, dentre todos que compõem o padrão original, retira-se os itens maiores remanescentes, em ordem não-crescente de tamanho, um de cada tipo por vez.

Para cada objeto do tipo k em estoque, um padrão de corte é construído e aquele que apresentar a menor perda é usado com frequência definida no Passo 9. Se não existir padrões com perda pequena, selecionamos o padrão com menor retalho. O padrão com menor retalho é escolhido para evitar soluções como da Figura 3.1 e - Solução 3.

4.1.4 Heurística Gulosa_A

O procedimento Guloso_A consiste em aplicar o procedimento Guloso para obter um padrão de corte e, logo após, a sobra é analisada. Se a sobra for aceitável (perda pequena ou retalho), o padrão é aceito, senão o item de maior comprimento no padrão de corte é retirado e a sobra é novamente analisada. Se o padrão ainda não for aceitável, então o segundo maior item é retirado. Este processo é repetido até que tenhamos uma sobra aceitável ou o padrão anulado. Se o padrão for anulado, escolhemos entre os padrões de corte originais, aquele que apresentar a menor sobra (esta é uma situação atípica que pode ocorrer, por exemplo, quando o estoque é formado apenas por retalhos).

A seguir, apresentamos os principais passos do algoritmo Guloso_A. Neste algoritmo, denotamos por e_k a quantidade de objetos do tipo $k, k = 1, \dots, K$, disponíveis no estoque e d_i a demanda do item tipo $i, i = 1, \dots, m$.

Algoritmo Guloso_A

$k = 1$

Passo 1: {*Início*}

Se $e_k = 0$ então $k = k + 1$, vá para o Passo 3;

Senão

aplique o procedimento Guloso para um objeto do tipo k e obtenha o padrão guloso, cujo vetor associado é α_k^{guloso} .

Passo 2: {Análise da sobra - padrão guloso}

Se a sobra for aceitável (perda pequena ou retalho) então

Início

padrão α_k^{guloso} é aceito {temporariamente armazenado na lista B};

Fim

Senão {alterações no padrão guloso}

Início

Enquanto a sobra não for aceitável, faça:

Início

Seja i = índice do item de maior comprimento no padrão α_k^{guloso} ;

Retire uma unidade do item i de α_k^{guloso} ;

Fim

Se $\alpha_k^{guloso} \neq 0$ então

Início

padrão α_k^{guloso} é aceito {temporariamente armazenado na lista B};

Fim

Senão

Início

padrão α_k^{guloso} obtido no Passo 1 é aceito { temporariamente armazenado na lista C};

Fim

$k = k + 1$, vá para o Passo 3.

Fim

Passo 3: {Determinação do melhor padrão de corte, frequência e atualização da demanda e do estoque}

Se $k = K + 1$ então

Início

Se a lista $B = \emptyset$ então

Início

Escolha entre os padrões de corte da lista C aquele que apresentar menor perda e armazene;

Fim

Senão

Início

Escolha entre os padrões de corte da lista B aquele que apresentar menor perda ou se não existir padrão com perda, aquele que apresentar menor o retalho e armazene;

Fim

Determine a frequência do padrão escolhido (respeitando-se as restrições de demanda e estoque) e armazene;

Atualize $d_i, i = 1, \dots, m$ e $e_k, k = 1, \dots, K$;

Se $d_i = 0, i = 1, \dots, m$ (solução factível encontrada) ou $e_k = 0, k = 1, \dots, K$ (demanda não atendida) então PARE;

Senão $k = 1$ e esvazie as listas B e C;

Fim

Vá para o Passo 1.

Em geral, nos procedimentos FFD_A e $Gulosa_A$, os últimos padrões são compostos por itens maiores, mais difíceis de serem combinados com os demais. Isto faz com que estes padrões (que devem combinar itens grandes) tenham sobras grandes, isto é, retalhos, algo não totalmente indesejável.

Estes dois procedimentos heurísticos fornecem soluções com perdas bastante reduzidas para o problema de aproveitamento de sobras (Seção 6.1), entretanto, muitos retalhos são mantidos no estoque. Para evitar esta situação, utilizamos a mesma estratégia da heurística FFD_A e desenvolvemos a heurística FFD_A^P que, além de considerar o aproveitamento de sobras, prioriza o corte de objetos não padronizados.

Para priorizar o corte de objetos não padronizados do estoque alteramos apenas a heurística FFD_A pelo fato desta heurística apresentar soluções com perdas baixas (assim como a heurística $Gulosa_A$) e gerar uma quantidade menor de retalhos comparada com a heurística $Gulosa_A$.

4.1.5 Heurística FFD_A^P

A heurística FFD_A^P , segue os mesmos passos da heurística FFD_A , porém, como além de minimizar a perda, desejamos reduzir a quantidade de objetos não padronizados disponíveis no estoque, consideramos inicialmente apenas os objetos não padronizados para corte. Quando todos estes objetos tiverem sido cortados, ou quando não for mais possível construir padrões de corte com sobras aceitáveis (perda pequena ou retalho) para esses objetos, consideramos então os objetos padronizados para corte com a finalidade de que toda demanda seja atendida. Observe que é possível atender toda a demanda apenas com o corte de objetos não padronizados.

A seguir, apresentamos os principais passos do algoritmo FFD_A^P .

Algoritmo FFD_A^P

Passo 1:

Aplique FFD_A considerando somente os objetos não padronizados.

Passo 2:

Se a demanda de todos os itens for atendida, PARE.

Senão

Aplique FFD_A com os objetos padronizados na demanda residual.

Além das heurísticas construtivas, heurísticas residuais também foram modificadas para resolver o problema de corte com sobras aproveitáveis. Antes de apresentarmos estas heurísticas, fazemos uma breve revisão de alguns conceitos e de algumas heurísticas residuais clássicas.

4.2 Heurísticas Residuais

Heurísticas Residuais encontram uma solução inteira para o problema de corte de estoque unidimensional, a partir de uma solução obtida com a relaxação linear do modelo matemático proposto por Gilmore e Gomory (1963):

$$\text{minimizar } f(x) = \sum_{k=1}^K \sum_{j=1}^{N_k} c_{jk} x_{jk} \quad (4.2)$$

sujeito a:

$$\sum_{k=1}^K \sum_{j=1}^{N_k} \alpha_{ijk} x_{jk} = d_i, \quad i = 1, \dots, m \quad (4.3)$$

$$\sum_{k=1}^{N_k} x_{jk} \leq e_k, \quad k = 1, \dots, K \quad (4.4)$$

$$x_{jk} \geq 0 \text{ e inteiro, } j = 1 \dots, N_k, \quad k = 1 \dots, K \quad (4.5)$$

em que

- c_{jk} é a sobra de material no padrão de corte j para o objeto do tipo k em estoque dada por $c_{jk} = L_k - \sum_{i=1}^m \ell_i \alpha_{ijk}$, $j = 1, \dots, N_k$, $k = 1, \dots, K$;
- x_{jk} é a variável de decisão que representa o número de objetos do tipo k cortados de acordo com o padrão j , $j = 1, \dots, N_k$, $k = 1, \dots, K$;
- $\alpha_{jk} = (\alpha_{1jk}, \alpha_{2jk}, \dots, \alpha_{mjk})$ é um padrão de corte em que α_{ijk} é o número de itens do tipo i no padrão de corte j para o objeto do tipo k , $i = 1, \dots, m$, $j = 1, \dots, N_k$, $k = 1, \dots, K$. No caso unidimensional, cada objeto do tipo k em estoque deve satisfazer:

$$\begin{cases} \ell_1 \alpha_{1jk} + \ell_2 \alpha_{2jk} + \dots + \ell_m \alpha_{mjk} \leq L_k \\ 0 \leq \alpha_{ijk} \leq d_i, \quad i = 1, \dots, m \text{ e } \alpha_{ijk} \text{ inteiro} \end{cases} \quad (4.6)$$

No modelo (4.2)-(4.5) a função objetivo (4.2) consiste em minimizar a perda de material. As restrições de igualdade em (4.3) garantem que as quantidades de itens produzidos sejam exatamente iguais as quantidades de itens demandados (qualquer peça cortada que não seja um item demandado é considerada sobra). As restrições (4.4) garantem que o número de objetos do tipo k cortados não exceda a disponibilidade e_k , $k = 1, \dots, K$ e por fim, as restrições (4.5) garantem que a repetição de cada objeto seja um número inteiro não-negativo.

Por simplicidade de notação, o modelo (4.2)-(4.5) pode ser escrito matricialmente como:

$$\begin{array}{ll} \text{minimizar} & f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\ \text{sujeito a :} & \left\{ \begin{array}{l} \mathbf{A}\mathbf{x} = \mathbf{d} \\ \mathbf{E}\mathbf{x} \leq \mathbf{e} \\ \mathbf{x} \geq \mathbf{0}, \text{ e inteiro} \end{array} \right. \end{array} \quad (4.7)$$

em que \mathbf{A} é a matriz dos padrões de corte em (4.3) e \mathbf{E} é a matriz de 0's e 1's na restrição (4.4).

A condição de integralidade sobre as variáveis x_{jk} dificulta computacionalmente a resolução dos problemas de corte na medida em que m cresce (já é bastante difícil quando m é da ordem de algumas poucas dezenas). Entretanto, Gilmore e Gomory (1961) propuseram relaxar esta condição de integralidade sobre as variáveis x_{jk} e resolver o problema resultante utilizando um procedimento de *geração de colunas*. A partir da solução ótima do problema relaxado, que geralmente não é inteira, determina-se uma solução inteira para o problema de corte de estoque original. Esta solução inteira pode ser determinada por procedimentos heurísticos que vêm sendo desenvolvidos por vários pesquisadores, como Stadtler (1990), Wäscher e Gau (1996), Poldi e Arenales (2009), entre outros. Alguns destes procedimentos heurísticos são apresentados nas seções (4.2.1) e (4.2.2).

Definição 4.1 *Seja \mathbf{x} uma solução ótima fracionária para a relaxação do problema (4.7) e seja \mathbf{y} um vetor de números inteiros não-negativos, próximos de \mathbf{x} tal que:*

$$\mathbf{A}\mathbf{y} \leq \mathbf{d} \quad (4.8)$$

$$\mathbf{E}\mathbf{y} \leq \mathbf{e} \quad (4.9)$$

o vetor \mathbf{y} , obtido de \mathbf{x} , é chamado de *solução inteira aproximada de \mathbf{x}* .

Uma possível maneira de obtermos uma solução inteira aproximada \mathbf{y} de \mathbf{x} é por um truncamento trivial:

$$\mathbf{y} = (\lfloor x_1 \rfloor, \lfloor x_2 \rfloor, \dots, \lfloor x_n \rfloor) \quad (4.10)$$

o qual satisfaz (4.8)-(4.9) uma vez que todos os coeficientes de \mathbf{A} e \mathbf{E} são não-negativos e \mathbf{x} satisfaz $\mathbf{A}\mathbf{x} = \mathbf{d}$ e $\mathbf{E}\mathbf{x} \leq \mathbf{e}$.

Definição 4.2 (*Problema Residual*)

Seja \mathbf{y} uma solução inteira aproximada de \mathbf{x} , $\mathbf{r} = \mathbf{d} - \mathbf{A}\mathbf{y}$ a demanda residual e $\mathbf{s} = \mathbf{e} - \mathbf{E}\mathbf{y}$ o estoque residual de objetos disponíveis. O problema residual é definido por (4.7), com $\mathbf{d} = \mathbf{r}$ e $\mathbf{e} = \mathbf{s}$.

Nas heurísticas residuais resolve-se a relaxação linear do problema (4.7) e obtém-se uma solução inteira aproximada. Então resolve-se a relaxação linear do problema residual (Definição 4.2) e obtém-se uma solução inteira aproximada e, assim sucessivamente, até que a demanda residual se anule ou a solução inteira aproximada do problema residual seja nula. Neste último caso, aplica-se algum método (heurístico ou exato) que resolve o problema residual final com poucos itens. A seguir, apresentamos a estrutura geral dessas heurísticas.

Algoritmo Residuais (Poldi e Arenales, 2009)

Passo 1: {Início}

Faça $\ell = 0$, $\mathbf{r}^0 = \mathbf{d}$ e $\mathbf{s}^0 = \mathbf{e}$;

Passo 2: {Determinação da solução ótima contínua}

Resolva o problema residual com $\mathbf{r} = \mathbf{r}^\ell$ e $\mathbf{s} = \mathbf{s}^\ell$;

Seja \mathbf{x}^ℓ a solução contínua (a técnica de geração de colunas é usada);

Se \mathbf{x}^ℓ é uma solução inteira, então PARE.

Passo 3: {Determinação da solução inteira aproximada}

Determine \mathbf{y}^ℓ , a solução inteira aproximada de \mathbf{x}^ℓ ;

Se \mathbf{y}^ℓ é um vetor nulo, então vá para o Passo 5.

Passo 4: {Atualização}

Determine a nova demanda residual

$$\mathbf{r}^{\ell+1} = \mathbf{r}^\ell - \mathbf{A}\mathbf{y}^\ell;$$

$$\mathbf{s}^{\ell+1} = \mathbf{s}^\ell - \mathbf{E}\mathbf{y}^\ell;$$

$$\ell = \ell + 1;$$

Vá para o Passo 2.

Passo 5:

Resolva o problema residual final com poucos itens por algum método, heurístico ou exato.

4.2.1 Heurísticas Residuais FFD, Gulosa, FFD_A e Gulosa_A

Para que o algoritmo residual (Seção 4.2) esteja totalmente definido, é necessário especificar como determinar \mathbf{y}^ℓ , solução inteira aproximada de \mathbf{x}^ℓ no Passo 3 e como resolver o problema residual no Passo 5. Denotamos heurísticas residuais FFD, Gulosa, FFD_A e Gulosa_A aquelas obtidas aplicando-se o algoritmo residual em que no Passo 3, a solução inteira aproximada é definida por (4.10) (ou seja, \mathbf{y}^ℓ é determinada por um truncamento trivial dado pelo inteiro inferior ao fracionário obtido) e, no Passo 5, se existir itens com demandas não atendidas, utiliza-se a heurística FFD, Gulosa, FFD_A ou Gulosa_A, respectivamente.

4.2.2 Heurísticas Residuais de Arredondamento Guloso (RAG)

Poldi e Arenales (2009) desenvolveram um procedimento de arredondamento guloso para obter uma solução inteira aproximada a partir de uma solução contínua \mathbf{x} no Passo 3 do algoritmo residual.

No procedimento de arredondamento guloso de Poldi e Arenales (2009), o Passo 3 do algoritmo residual é dividido em duas partes: Passo de *Pré-processamento* e Passo de *Arredondamento*. Para estas heurísticas, referenciadas como heurísticas RAG, o Passo 5 do algoritmo residual não ocorre, pois todas as demandas são satisfeitas já que a frequência de pelo menos um padrão de corte é arredondada para o inteiro superior ao fracionário obtido e, desta forma, \mathbf{y}^ℓ nunca será nulo no Passo 3.

No Passo de *Pré-processamento* ordena-se os padrões de corte da solução contínua \mathbf{x}^ℓ (obtidas no Passo 2 do algoritmo residual) segundo um dos seguintes critérios: RAG - versão 1, RAG - versão 2 ou RAG - versão 3, descritos a seguir. Por simplicidade de notação, considere T padrões de corte, com frequências não nulas, obtidos no Passo 2 do algoritmo residual (Seção 4.2), enumerados por $1, 2, \dots, T$ e k_j o objeto associado ao padrão de corte $j, j = 1, \dots, T$.

- RAG 1: os padrões de corte são ordenados de acordo com os valores não-crescentes das frequências, ou seja, $x_{1k_1} \geq x_{2k_2} \geq \dots \geq x_{Tk_T}$;
- RAG 2: os padrões de corte são ordenados de acordo com os valores não-decrescentes

de suas sobras, ou seja, $c_{1k_1} \leq c_{2k_2} \leq \dots \leq c_{Tk_T}$;

- RAG 3: seja $f_{jk} = x_{jk} - \lfloor x_{jk} \rfloor$ a parte fracionária de x_{jk} . Os padrões de corte são ordenados de acordo com os valores não-crescentes dessas partes fracionárias, ou seja, $f_{1k_1} \geq f_{2k_2} \geq \dots \geq f_{Tk_T}$.

No Passo de *Arredondamento*, iniciamos com o primeiro padrão de corte, conforme uma das ordenações anteriores, e sua frequência é arredondada para o inteiro superior ao fracionário obtido, ou seja, $y_{1k_1} = \lceil x_{1k_1} \rceil$. As demais frequências são fixadas em 0 ($y_{jk_j} = 0, j = 2, \dots, T$) e as condições (4.8) e (4.9) são testadas. Caso haja violação, y_{1k_1} é reduzido sucessivamente de uma unidade ($y_{1k_1} = y_{1k_1} - 1$), até que tenhamos uma solução inteira aproximada. O valor de y_{1k_1} é fixado e o procedimento é repetido com o segundo padrão de corte e, assim, determinamos uma nova solução inteira aproximada: y_{1k_1}, y_{2k_2} e $y_{jk_j} = 0, j = 3, \dots, T$. Este procedimento é repetido com todos os padrões de corte.

Para resolver o problema de corte de estoque com sobras de material aproveitáveis, adaptamos este arredondamento guloso para obter as heurísticas RAG_A - versões 1, 2 e 3. Nestes procedimentos heurísticos, os objetos em estoque têm as mesmas chances de serem utilizados, ou seja, não priorizamos o corte dos objetos não padronizados (Seção 4.2.4).

4.2.3 Heurísticas RAG_A

As heurísticas RAG_A utilizam um limitante superior para o percentual máximo da perda, calculada a partir da solução inteira aproximada obtida utilizando uma das versões da heurística RAG (nas heurísticas construtivas esse limitante é fornecido pelo usuário, para objetos padronizados e não padronizados).

Cálculo do Percentual Máximo Aceitável de Perda

1. Considere os padrões de corte obtidos por uma das versões da heurística RAG (Seção 4.2.2).
2. Determine a sobra total, desconsiderando os retalhos (isto é, sobras de material no padrão de corte superiores ou iguais a δ):

$$\sigma = \sum_{j/c_{jk_j} < \delta} c_{jk_j} x_{jk_j}$$

3. Determine o comprimento total dos objetos cortados com sobras (os objetos cortados com retalhos são desconsiderados):

$$\gamma = \sum_{j/c_{jk_j} < \delta} L_{jk_j} x_{jk_j}$$

4. O percentual máximo para uma perda aceitável (perda pequena) é dado por:

$$\rho = \frac{\sigma}{\gamma}$$

Seja $\lambda_{jk_j} = \frac{c_{jk_j}}{L_{k_j}}$ a fração perdida no padrão $j, j = 1, \dots, T$ do objeto k_j . Os principais passos das heurísticas RAG_A são:

Algoritmo RAG_A

Passo 1: {Início}

Determine uma solução inteira aproximada usando o procedimento de arredondamento guloso segundo o critério RAG (versões 1, 2 ou 3).

Passo 2: {Análise das perdas geradas pelos padrões}

Analise a perda de todos os padrões gerados no Passo 1;

Se a perda for *aceitável*, ou seja, $\lambda_{jk_j} \leq \rho$ (obtido no Cálculo do Percentual Máximo Aceitável de Perda) então

padrão j para o objeto k_j é aceito e armazenado com frequência obtida no Passo 1;

Senão

padrão j é rejeitado e a demanda dos itens no padrão j e o estoque dos objetos tipo k_j são atualizados.

Passo 3: {Problema Residual}

Se não houver padrão de corte rejeitado no Passo 2, PARE, senão aplique o procedimento FFD_A para resolver o problema residual formado pelos itens contidos nos padrões de corte rejeitados no Passo 2 e objetos restantes.

Outros procedimentos para resolução do problema residual (Passo 3) podem ser utilizados gerando outras heurísticas. Entretanto, sugerimos o procedimento FFD_A pois,

como veremos nos testes computacionais realizados (Seção 6.1), esta heurística gera perdas menores quando comparada com as heurísticas FFD e Gulosa e uma menor quantidade de objetos cortados com retalhos comparada com a heurística Gulosa_A.

Para priorizar o corte de objetos não padronizados disponíveis no estoque, propomos algumas alterações para a heurística RAG_A - versão 2. Das três versões desenvolvidas da heurística RAG_A alteramos apenas a versão 2 pelo fato desta heurística priorizar padrões de corte com perdas mais baixas durante o processo de arredondamento e esta técnica, conseqüentemente, nos leva a maior utilização dos padrões de corte gerados. A nova versão desta heurística é denominada de RAG_A^P.

4.2.4 Heurística RAG_A^P

Para resolver o problema de corte de estoque com sobras de material aproveitáveis, três versões da heurística RAG_A foram apresentadas. Estas heurísticas apresentam boas soluções para o problema de corte de estoque com sobras aproveitáveis, reduzindo a perda significativamente. Entretanto, é necessário administrar um grande número de retalhos visto que somente as perdas são otimizadas por estas heurísticas.

Diferentemente das heurísticas RAG_A, a heurística RAG_A^P utiliza como limite para classificar uma sobra como *perda pequena* nos objetos não padronizados o parâmetro β (Seção 3.2) que é um dado de entrada fornecido pelo usuário.

A estrutura da heurística RAG_A^P é a mesma da heurística RAG_A, porém, como a prioridade é reduzir a sobra e o número de objetos não padronizados do estoque, na função objetivo do modelo (4.2)-(4.5) consideramos c_{jk} o coeficiente de custo associado a cada objeto do tipo k em estoque, $k = 1, \dots, K$. Com este coeficiente, custos mais baixos são associados aos objetos não padronizados (objetos $k = \bar{k} + 1, \dots, K$) como segue,

$$c_{jk} = \begin{cases} cL_k, & k = 1, \dots, \bar{k} \\ \gamma(cL_k), & k = \bar{k} + 1, \dots, K \end{cases} \quad (4.11)$$

em que c é um custo por unidade de comprimento e γ é um desconto que incentiva o uso de objetos não padronizados, $0 < \gamma < 1$.

Além de atribuímos custos diferenciados para os objetos em estoque, na heurística

RAG_A^P o problema residual formado pelos padrões de corte rejeitados é resolvido pela heurística FFD_A^P que também prioriza o corte de objetos não padronizados.

Utilizando os parâmetros θ e β definidos na Seção (3.2) para classificar uma sobra como *perda pequena* e considerando λ_{jk_j} a sobra gerada no padrão de corte j para o objeto k_j , $j = 1, \dots, T$, em que T é a quantidade de padrões de corte com frequências não nulas, apresentamos os principais passos da heurística RAG_A^P .

Algoritmo RAG_A^P

Passo 1: {Início}

Determine uma solução inteira aproximada para o problema (4.2)-(4.5), com os coeficientes de custo conforme (4.11), utilizando o procedimento de arredondamento guloso RAG - versão 2 (Poldi e Arenales, 2009) e obtenha T padrões de corte com frequências positivas, em que o padrão j está associado ao objeto k_j , $j = 1, \dots, T$;

Passo 2: {Análise das perdas geradas pelos padrões}

Para cada padrão de corte j gerado no Passo 1, se a sobra for aceitável, isto é, $\lambda_{jk_j} \leq \theta L_{k_j}$, $k_j \in \{1, 2, \dots, \bar{k}\}$ ou $\lambda_{jk_j} \leq \beta$, $k_j \in \{\bar{k} + 1, \dots, K\}$, então o padrão de corte j para o objeto k_j é aceito e armazenado com a frequência obtida no Passo 1;

Senão

o padrão de corte j é rejeitado e a demanda e o estoque são atualizados.

Passo 3: {Problema Residual}

Se não houver padrão de corte rejeitado no Passo 2, PARE, senão aplique o procedimento FFD_A^P para resolver o problema residual formado pelos itens dos padrões de corte rejeitados no Passo 2 e objetos remanescentes do estoque.

A seguir, apresentamos um exemplo no qual exibimos a solução dos procedimentos heurísticos clássicos descritos e a solução dos procedimentos heurísticos que consideram o aproveitamento de sobras.

4.3 Exemplo

Para exemplificar os procedimentos heurísticos apresentados, suponha que temos um estoque composto por $k = 5$ tipos de objetos, sendo L_1 e L_2 objetos padronizados

e $L_k, k = 3, 4, 5$, objetos não padronizados, ou seja, retalhos. Os comprimentos destes objetos, assim como suas disponibilidades $e_k, k = 1, \dots, 5$, estão na Tabela 4.1.

Tabela 4.1: Dados dos objetos em estoque

| Objetos | Comprimento | estoque |
|---------|-------------|---------|
| 1 | 1000 | 100 |
| 2 | 1100 | 100 |
| 3 | 47 | 5 |
| 4 | 277 | 9 |
| 5 | 492 | 4 |

Para o procedimento heurístico RAG_A^P , os objetos não padronizados do estoque (objetos 3, 4 e 5) possuem um custo associado que é calculado a partir de (4.11) com $\gamma = 0,9$. Os demais procedimentos, por não priorizarem o corte dos objetos não padronizados não possuem qualquer custo associado.

As demandas a serem atendidas são compostas por $m = 10$ tipos de itens cujos dados estão na Tabela 4.2.

Tabela 4.2: Dados dos itens a serem produzidos

| Itens | Comprimento | demanda |
|-------|-------------|---------|
| 1 | 81 | 85 |
| 2 | 70 | 304 |
| 3 | 58 | 172 |
| 4 | 60 | 160 |
| 5 | 49 | 78 |
| 6 | 37 | 463 |
| 7 | 26 | 382 |
| 8 | 27 | 276 |
| 9 | 16 | 484 |
| 10 | 8 | 42 |

Para classificar uma sobra como *perda pequena* nos procedimentos heurísticos que não priorizam o corte de objetos não padronizados, definimos os parâmetros $\theta = 0,005$ e $\beta = 0,05$ para os objetos padronizados e não padronizados, respectivamente. Para a heurística RAG_A^P , definimos o parâmetro $\theta = 0,005$ e $\beta = 5$ para os objetos padronizados e não padronizados, respectivamente. Como limitante para o retalho, consideramos $\delta = 43$ (média dos comprimentos dos itens demandados).

A solução dos procedimentos heurísticos são apresentadas a seguir. Para as heurísticas residuais de arredondamento guloso (Tabela 4.4) não apresentamos as soluções

das heurísticas RAG 1 e 3 e RAG_A 1 e 3, pelo fato destas heurísticas gerarem as mesmas soluções que as heurísticas RAG 2 e RAG_A 2. A solução da heurística FFD_A^P também não é apresentada pois, embora esta heurística apresente soluções com perdas muito baixas e utilize todos os retalhos do estoque, gera uma grande de novos retalhos quando a demanda é composta por vários itens. Desta forma, a heurística FFD_A^P é utilizada apenas para resolver o problema residual da heurística RAG_A^P , quando a demanda dos itens é bem reduzida.

Nas Tabelas 4.3 e 4.4 a seguir, (Perda) representa a perda gerada no plano de corte, (Retalho) é o comprimento do retalho gerado, (Ret Gerado) representa a quantidade de retalhos gerados, (Ret Usado) é a quantidade de retalhos utilizada do estoque, (Comp Pad) e (Comp Ret) representam o comprimento total dos objetos padronizados e não padronizados utilizados, respectivamente.

Tabela 4.3: Solução das heurísticas construtivas

| | Heurísticas Construtivas | | | |
|------------|--------------------------|---------|--------|------------|
| | FFD | FFD_A | Gulosa | $Gulosa_A$ |
| Perda | 145 | 16 | 30 | 0 |
| Retalho | 232 | 157 | 0 | 119 |
| Ret Gerado | 1 | 1 | 0 | 1 |
| Ret Usado | 5 | 3 | 4 | 1 |
| Comp Pad | 94300 | 93500 | 94000 | 94000 |
| Comp Ret | 235 | 831 | 188 | 277 |

Tabela 4.4: Solução das heurísticas residuais

| | Heurísticas Residuais | | | | | | |
|------------|-----------------------|---------|--------|------------|-------|-----------|-----------|
| | FFD | FFD_A | Gulosa | $Gulosa_A$ | RAG 2 | RAG_A 2 | RAG_A^P |
| Perda | 37 | 12 | 20 | 0 | 116 | 14 | 10 |
| Retalho | 0 | 176 | 0 | 250 | 0 | 159 | 328 |
| Ret Gerado | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| Ret Usado | 5 | 3 | 4 | 4 | 7 | 3 | 18 |
| Comp Pad | 93500 | 93300 | 93300 | 93300 | 93500 | 93500 | 89800 |
| Comp Ret | 695 | 1046 | 878 | 1108 | 774 | 831 | 4696 |

Pelas tabelas 4.3 e 4.4 observamos que as heurísticas de aproveitamento geram perdas mais baixas que suas respectivas versões clássicas. Com relação aos retalhos, temos que, com exceção da heurística RAG_A^P que utiliza todos os objetos não padronizados

do estoque devido aos descontos atribuídos a eles, as demais heurísticas utilizam uma quantidade muito baixa destes objetos. Este fator tem como consequência o acúmulo de objetos não padronizados no estoque que podem acabar como sucatas caso a empresa não os utilize após um determinado período de tempo.

A redução dos objetos não padronizados do estoque, além de facilitar algumas atividades operacionais, também permite a redução do uso de objetos padronizados para atender as demandas, como observamos na heurística RAG_A^P .

No Capítulo 5 a seguir, apresentamos os procedimentos heurísticos desenvolvidos para resolver o problema de corte bidimensional com sobras aproveitáveis.

Capítulo 5

Métodos de solução para o problema de corte bidimensional com sobras aproveitáveis

Para resolver o problema de corte bidimensional com sobras aproveitáveis, algumas alterações foram realizadas na abordagem grafo E/OU (Morabito (1989), Vianna, 2000), pois se trata de uma estratégia com bastante flexibilidade para resolver problemas de corte quando duas dimensões são consideradas. Além das alterações no grafo E/OU, também apresentamos alterações realizadas em alguns procedimentos heurísticos (Poldi e Arenales, 2009) para a obtenção de soluções inteiras.

A seguir, apresentamos uma breve revisão da abordagem grafo E/OU e, em seguida, as alterações realizadas para resolver o problema de aproveitamento de sobras.

5.1 Abordagem grafo E/OU

A abordagem em grafo E/OU para resolução de problemas de corte foi inicialmente proposta por Morabito (1989) para problemas de corte bidimensionais, guilhotinados, irrestritos e não estagiados. Esta abordagem consiste em representar todos os padrões de corte como um caminho completo em um grafo e enumerá-los implicitamente com a finalidade de encontrar uma solução ótima.

5.1.1 Corte Guilhotinado

O corte é chamado guilhotinado ortogonal, ou simplesmente guilhotinado, quando aplicado em uma placa, produz dois novos retângulos, chamados de *placas intermediárias*, as quais podem ser cortadas novamente. Um padrão de corte é do tipo guilhotinado se for obtido por cortes guilhotinados sucessivos. A Figura 5.1 mostra um corte e um padrão de corte guilhotinado.

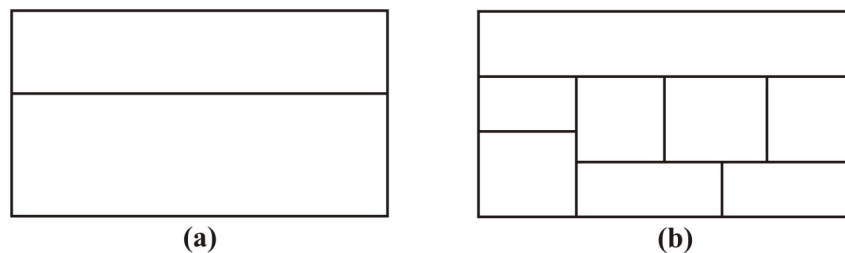


Figura 5.1: (a) Corte guilhotinado horizontal, (b) Padrão de corte guilhotinado.

Um padrão de corte pode ser organizado em estágios. Em cada estágio, cortes paralelos são aplicados em uma placa e no estágio seguinte, cortes ortogonais aos do estágio anterior são aplicados nas placas obtidas no estágio anterior. Se o número de estágios é limitado por k , dizemos que o padrão de corte é (k -estagiado).

A Figura 5.2 mostra um exemplo de padrão de corte guilhotinado 2-estágios.

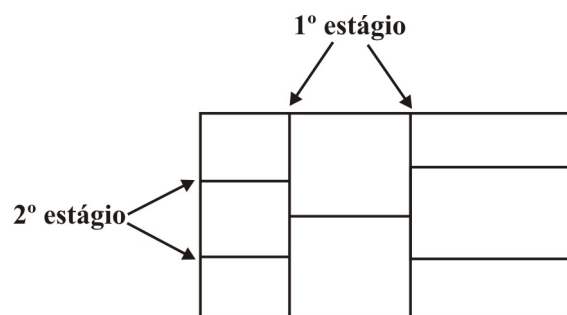


Figura 5.2: Padrão de corte guilhotinado 2-estágios.

Uma placa N (original ou intermediária), tem comprimento L_N e largura W_N e é denotada por $N = (L_N, W_N)$. O conjunto de possíveis itens que pode ser cortados de

$N = (L_N, W_N)$ é dado por:

$$M(N) = \{i : \ell_i \leq L_N \text{ e } w_i \leq W_N, i = 1, \dots, m\}.$$

Hertz (1972) mostrou que, sem perda de generalidade, as posições dos cortes guilhotinados podem ser combinações lineares não-negativas das dimensões dos itens. Desta forma, todo corte vertical na posição x pode ser tomado do conjunto:

$$X = \left\{ x : x = \sum_{i=1}^m \alpha_i \ell_i, \quad \ell_0 \leq x \leq L - \ell_0, \quad \alpha_i \geq 0 \text{ e inteiro}, i = 1, \dots, m \right\}$$

em que $\ell_0 = \min\{\ell_i, i = 1, \dots, m\}$.

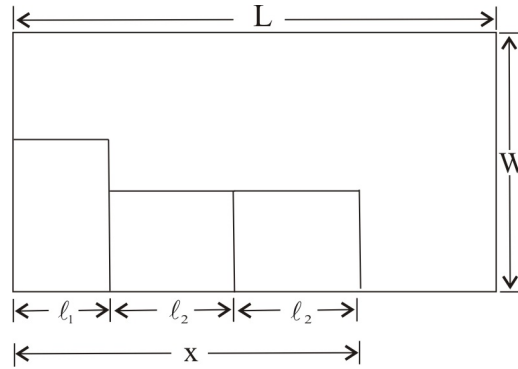


Figura 5.3: Combinação linear dos comprimentos ($x = \ell_1 + 2\ell_2$).

Analogamente, o conjunto Y define os cortes horizontais:

$$Y = \left\{ y : y = \sum_{i=1}^m \beta_i w_i, \quad w_0 \leq y \leq W - w_0, \quad \beta_i \geq 0 \text{ e inteiro}, i = 1, \dots, m \right\}$$

em que $w_0 = \min\{w_i, i = 1, \dots, m\}$.

Os conjuntos X e Y são chamados de *conjuntos de discretizações*. Os conjuntos de discretizações para uma placa intermediária $N = (L_N, W_N)$ são dados por:

$$X(N) = \left\{ x : x = \sum_{i \in M(N)} \alpha_i \ell_i, \quad \ell_0 \leq x \leq L_N - \ell_0, \quad \alpha_i \geq 0 \text{ e inteiro}, i \in M(N) \right\}$$

$$Y(N) = \left\{ y : y = \sum_{i \in M(N)} \beta_i w_i, \quad w_0 \leq y \leq W_N - w_0, \quad \beta_i \geq 0 \text{ e inteiro}, i \in M(N) \right\}$$

os quais podem ser construídos por uma fórmula recursiva dada por Christofides e Whittlock (1972) ou pela sua versão revisada, apresentada em Morabito e Arenales (1996).

Desta forma, dada uma placa intermediária $N = (L_N, W_N)$, um corte vertical na posição $x \in X(N)$ produz dois novos nós: $N_1 = (x, W_N)$ e $N_2 = (L_N - x, W_N)$. O mesmo ocorre para os cortes horizontais $y \in Y(N)$.

Considerando somente os cortes em X e Y , temos que o número de placas intermediárias geradas é finito, porém, um número muito grande.

5.1.2 Grafo E/OU

Um grafo E/OU, denotado por $G = (V, E)$, é um tipo especial de grafo, em que V é o conjunto de nós (ou vértices) e $E = \{e_1, \dots, e_s\}$ é o conjunto de arcos direcionados. Cada arco e_{ju} do grafo, liga um nó j a um conjunto S_u de nós:

$$e_{ju} = (j, S_u), j \in V, S_u \subseteq V.$$

Os nós em S_u são chamados *sucessores* de j e j , é chamado *predecessor* dos nós em S_u . Para representar um corte guilhotinado, S_u consiste de um par $\{p, q\}$ de nós, ou um conjunto unitário $\{p\}$.

Quando seguimos um caminho pelo grafo, podemos escolher entre os vários arcos que emergem de um nó (arcos-OU), entretanto, neste caminho, temos que seguir ambas ramificações do arco escolhido (arco-E). Na Figura 5.4 - (a), as linhas pontilhadas indicam um possível caminho no grafo E/OU.

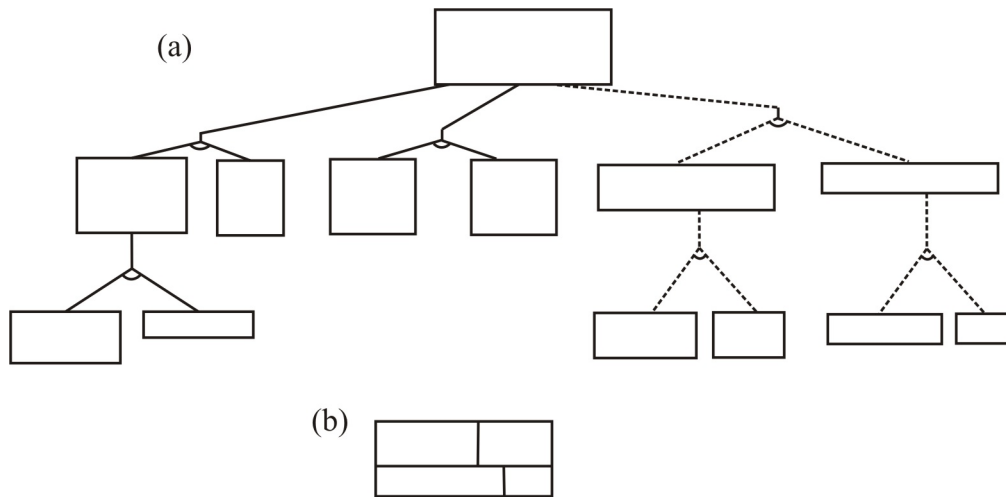


Figura 5.4: (a) Grafo E/OU com um caminho particular em destaque; (b) Padrão de corte correspondente.

Este tipo de grafo fornece uma ferramenta apropriada para representar um processo de corte, sendo que cada nó representa uma placa do estoque ou uma placa intermediária e, cada arco $e_{ju} = (j, \{p, q\})$, representa um corte guilhotinado que divide uma placa/nó j (nó inicial de e_{ju}) em um par de novas placas $\{p, q\}$ (nós finais de e_{ju}).

Os padrões de corte são gerados examinando-se todas as possibilidades de corte, entretanto, em algum ponto durante o processo de corte, nenhum corte adicional pode ser feito em uma placa particular (ou pelas dimensões desta placa indicar que uma perda deve ser considerada, ou itens com dimensões exatas foram gerados, ou porque é conhecida a solução ótima desta placa). Esta situação é representada no grafo E/OU introduzindo um arco, chamado de 0-corte (ou arco-0) para cada nó j , isto é, um arco $e_{ju} = (j, p)$ que copia exatamente a placa do nó j para o nó p . Se este arco é escolhido no caminho representado pelo grafo E/OU nenhum outro corte pode ser feito no nó p , o qual passa a ser chamado de *nó final* do caminho. Cada nó final possui um determinado valor associado, que é igual a zero no caso em que o nó representa uma sobra. Se o nó final representa um item i específico, então seu valor é idêntico ao valor de i . No caso em que o nó final representa uma placa para a qual o padrão de corte ótimo é conhecido, então seu valor é idêntico ao valor deste padrão.

Qualquer padrão de corte para um objeto (placa em estoque) pode ser determinado

pelo grafo E/OU como segue: Iniciando do nó raiz (placa em estoque), escolha um e somente um arco (arco E ou 0-corte) e para cada nó apontado por este arco escolha novamente um e somente um arco e assim por diante, até que todos os nós visitados sejam *nós finais*. Esta seqüência é chamada de *caminho completo* do grafo E/OU e corresponde a um padrão de corte. As linhas pontilhadas na Figura (5.4 - a) indica um caminho completo o qual corresponde a um padrão de corte (Figura 5.4 - b).

O valor de um caminho completo gerado pelo grafo E/OU (ou equivalentemente, padrão de corte) é dado pela soma dos valores de todos os nós finais. Desta forma, determinar um padrão de corte consiste em determinar o caminho completo mais valioso no grafo E/OU.

A Figura 5.6 a seguir ilustra três caminhos completos para uma placa A em estoque e, portanto, três possíveis padrões de corte.

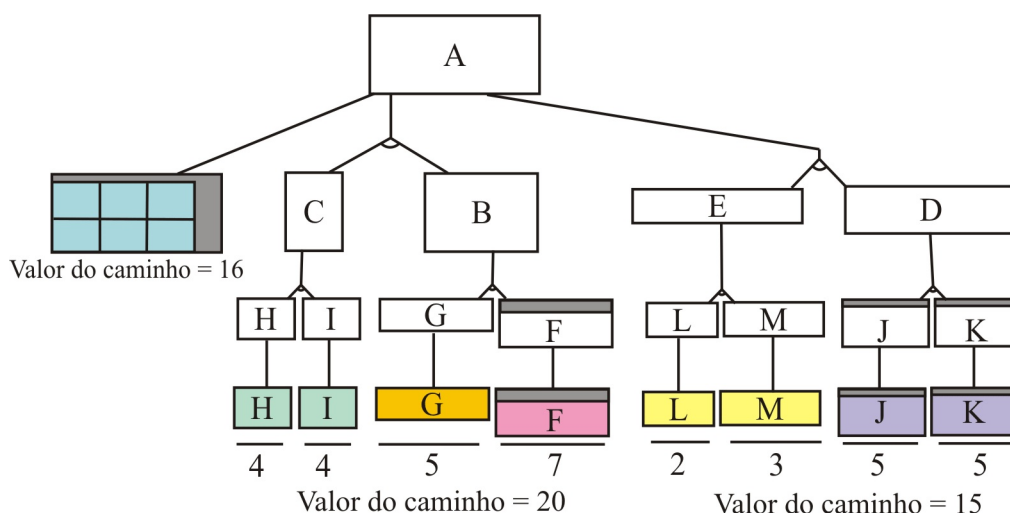


Figura 5.5: Grafo E/OU representando padrões de corte alternativos.

Na Figura 5.5 três arcos-OU emergem da raiz, indicando três alternativas diferentes para se obter padrões de corte. Nesta representação, o caminho com valor 20, por ser o mais valioso, fornece o padrão de corte para a placa A. O caminho com valor 16 representa um padrão de corte homogêneo (fácil de ser computado) e é utilizado como limitante inferior (Seção 5.1.3) para as demais ramificações.

5.1.3 Limitantes inferior e superior

Para descrever o método de enumeração implícita do caminho completo (isto é, do padrão de corte), definimos limitantes inferiores e superiores para o valor do padrão de corte ótimo para uma placa $N = (L_N, W_N)$.

A determinação de um limitante inferior, $LI(N)$, para o valor de uma função objetivo relacionada a uma placa $N = (L_N, W_N)$ pode ser baseada em padrões de corte homogêneos, que são compostos apenas por itens de um mesmo tipo.

Para uma placa $N = (L_N, W_N)$, o número máximo de vezes que um item k , (ℓ_k, w_k) , aparece no padrão de corte homogêneo é $\left(\left\lfloor \frac{L_N}{\ell_k} \right\rfloor \cdot \left\lfloor \frac{W_N}{w_k} \right\rfloor \right)$, sendo o correspondente valor da função objetivo dado por $\left(v_k \cdot \left\lfloor \frac{L_N}{\ell_k} \right\rfloor \cdot \left\lfloor \frac{W_N}{w_k} \right\rfloor \right)$. Desta forma, o melhor padrão de corte homogêneo que fornece um limitante inferior para o valor da função objetivo para a placa $N = (L_N, W_N)$ é:

$$LI(N) = LI(L_N, W_N) = \max \left\{ v_i \cdot \left\lfloor \frac{L_N}{\ell_i} \right\rfloor \cdot \left\lfloor \frac{W_N}{w_i} \right\rfloor : i \in M(N) \right\} \quad (5.1)$$

Quando necessário, o padrão de corte homogêneo também pode considerar a rotação de itens (rotação de 90°).

Além do limitante inferior, durante o processo de busca no grafo E/OU também utilizamos limitantes superiores, $LS(N)$. A área fornece um modo direto para computarmos um limitante superior para o valor da função objetivo, ou seja,

$$LS(N) = LS(L_N, W_N) = \max \sum_{i \in M(N)} a_i v_i \quad (5.2)$$

Sujeito a:

$$\sum_{i \in M(N)} a_i (\ell_i w_i) \leq L_N \cdot W_N \quad (\text{utilização da área}) \quad (5.3)$$

$$a_i \geq 0, \quad i \in M(N). \quad (5.4)$$

No modelo (5.2) - (5.4), a_i representa o número de itens do tipo i , $i = 1, \dots, m$, presentes no padrão de corte.

Trivialmente, o valor ótimo da função objetivo do problema (5.2)-(5.4) é obtido por:

$$LS(L_N, W_N) = L_N \cdot W_N \cdot \max \left\{ \frac{v_i}{\ell_i w_i} : i \in M(N) \right\} \quad (5.5)$$

A determinação do limitante inferior e superior requer pouco tempo computacional, sendo que o número de operações realizadas depende linearmente do número de tipos de itens pequenos que devem ser cortados.

5.1.4 Algoritmo Branch & Bound

Para descrevermos o algoritmo de enumeração implícita (*Branch & Bound*), considere $LI(N)$ o limitante inferior e $LS(N)$ o limitante superior para o valor da função objetivo do nó/placa N e $V(N)$ o melhor valor conhecido da função objetivo atual para o nó N , sendo que $LI(N) \leq V(N) \leq LS(N)$.

Além disso, seja S o conjunto de nós (placa original ou intermediária) a ser serem cortados. O procedimento *Branch & Bound* que pode ser utilizado para determinar a solução ótima do problema de corte bidimensional é apresentado a seguir.

Algoritmo Branch & Bound

Passo 1: {Início}

$$S = \{(L, W)\}$$

Passo 2: {Determinação e verificação da solução}

Escolha um nó $N = (L_N, W_N)$ de S e apague ele de S . Faça $V(N) = LI(N)$.

Para todos os possíveis sucessores, N_1 e N_2 , que podem resultar de um corte vertical em N escolhido do conjunto de discretização $X(N)$ ou de um corte horizontal em N escolhido do conjunto de discretização $Y(N)$ faça:

- a) Se N_1 representa uma perda, ou seja, $M(N_1) = \emptyset$, então faça de N_1 um nó final, caso contrário, inclua N_1 em S . Repita esse passo para N_2 .
- b) Se $V(N) < LI(N_1) + LI(N_2)$, ou seja, é possível melhorar o valor da função objetivo por este corte, então atualize $V(N) = LI(N_1) + LI(N_2)$. Também atualize recursivamente os melhores valores conhecidos das funções objetivo de todos os predecessores de N , até a placa original (nó raiz).

- c) Se $V(N) \geq LS(N_1) + LS(N_2)$, ou seja, o corte pode não resultar em um valor melhor para a função objetivo, então remova N_1 e N_2 de S .

Passo 3:

Se $S \neq \emptyset$, vá para o passo 2.

Passo 4:

PARE! O melhor valor atual conhecido para o nó raiz é o valor ótimo da função objetivo. O correspondente padrão de corte pode ser determinado identificando o caminho que fornece este valor para a função objetivo.

Como existem diferentes maneiras de percorrer um grafo, para reduzir o espaço de busca da melhor solução no grafo E/OU, heurísticas podem ser aplicadas de modo a evitar caminhos não promissores. Algumas dessas heurísticas são descritas a seguir.

5.1.5 Heurísticas

O tempo computacional do algoritmo *Branch & Bound* depende do número de nós que devem ser considerados, o qual pode ser enorme. Desta forma, Morabito *et al.* (1992) desenvolveram várias heurísticas, as quais são modificações de procedimentos originais que reduzem consideravelmente o tempo computacional e, mesmo assim, levam a boas soluções.

Heurística 1: (Uso de cortes promissores)

Um corte é considerado promissor se a soma dos limitantes superiores das placas resultantes N_1 e N_2 é relativamente maior que o melhor valor atual conhecido de N , ou seja, $LS(N_1) + LS(N_2) > (1 + \alpha)V(N)$ e se a soma dos limitantes inferiores dos sucessores é ligeiramente menor ou mesmo maior que o limitante inferior de N , ou seja, $LI(N_1) + LI(N_2) > (1 - \beta)LI(N)$, em que α e β são parâmetros escolhidos empiricamente.

Heurística 2: (Heurística de estratégia de busca)

Morabito *et al.* (1992) combinaram duas estratégias básicas para evitar caminhos não promissores no grafo E/OU: busca em profundidade e hill-climbing. Desta forma, escolheram arbitrariamente um limite de profundidade (um número máximo de cortes sucessivos, ou o comprimento do caminho) e computaram do nó raiz (placa inicial) o

melhor caminho completo (exato ou heurístico) não excedendo este limite e descartando todos os outros caminhos (uma busca hill-climbing pura seria obtida para a profundidade limite de 1). Então, neste melhor caminho completo, para uma determinada profundidade limite, todos os nós finais são considerados raízes e escolhidos para investigação e, assim por diante. Para detalhes, veja Morabito *et al.* (1992) ou Arenales e Morabito (1995).

Após esta revisão da abordagem em grafo E/OU, apresentamos as alterações realizadas neste método para resolver o problema de corte com sobras aproveitáveis.

5.2 Abordagem grafo E/OU para o aproveitamento de sobras

Para resolver o problema de aproveitamento de sobras, alteramos os padrões de corte gerados pela abordagem grafo E/OU com a finalidade de evitar soluções com perdas elevadas, porém, com um número reduzido de retalhos.

Para um melhor entendimento das modificações realizadas no grafo E/OU, suponha que para uma determinada placa padronizada do estoque, temos um padrão de corte, isto é, um caminho completo composto por três nós finais, ilustrado na Figura 5.6:

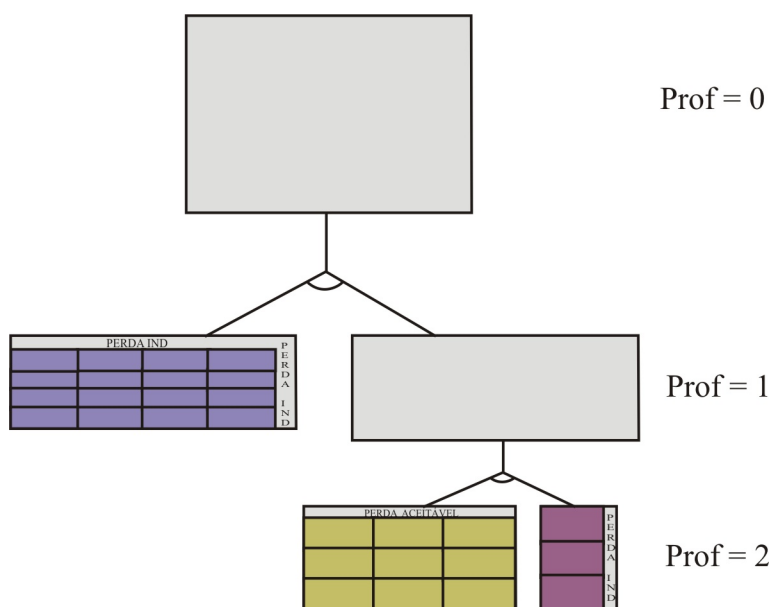


Figura 5.6: Solução gerada pelo Grafo E/OU e análise dos nós finais.

Na Figura 5.6, observamos que o padrão de corte possui um nó final com perda aceitável (satisfaz (3.2) ou (3.3)) e nós finais com perdas indesejáveis (perda ind), ou seja, o padrão de corte possui sobras que não satisfazem o critério (3.1) nem os critérios (3.2) ou (3.3).

De acordo com (3.2), para classificarmos uma sobra em um nó final como perda aceitável (perda pequena), $A_s < \xi L_k W_k, k = 1, \dots, \bar{k}$. Entretanto, nesta condição, o parâmetro ξ , ($0 < \xi < 1$), é estimado considerando 2 casos:

$$\xi = \frac{\psi_k}{2^{prof}}, \quad \text{se } prof = 1, 2$$

$$\xi = \frac{\psi_k}{4}, \quad \text{se } prof > 2$$

em que $prof$ é a profundidade de busca no grafo E/OU e $\psi_k, k = 1, \dots, \bar{k}$, é a porcentagem de perda mínima admissível para cada placa k em estoque. O valor de ψ_k é determinado a partir da porcentagem de perda obtida da solução do problema de corte para a placa k , considerando todos os itens demandados. Entretanto, se esta porcentagem for nula ou estiver muito próxima de zero, um valor arbitrário é estabelecido para ψ_k de modo que $0 < \psi_k < 1$, por exemplo, $\psi_k = \min\{\ell_i w_i, i = 1, \dots, m\} / L_k W_k$.

Na restrição (3.2) limitamos a análise da sobra pela profundidade da busca no grafo E/OU pois, à medida que a profundidade da busca aumenta ($prof > 2$), os limites para a perda tornam-se muito pequenos e desta forma, bons padrões de corte poderiam ser rejeitados.

Para o problema de corte de estoque com sobras aproveitáveis, como objetivamos perda mínima, as perdas indesejáveis na Figura 5.6 devem ser eliminadas. Entretanto, além da perda mínima, a quantidade de retalhos que retorna ao estoque também deve ser reduzida, caso contrário, os padrões de corte poderiam ser gerados sem perdas, porém com uma grande quantidade de retalhos, tornando a solução operacionalmente inviável. Desta forma, os retalhos foram classificados em dois grupos: *retalho natural* e *retalho artificial*. O *retalho natural* satisfaz a condição (3.1) sem que alterações sejam realizadas no padrão de corte, enquanto que o *retalho artificial*, é gerado a partir de alterações realizadas nos padrões de corte até que (3.1) seja satisfeita.

Considerando esta classificação para os retalhos, admitimos que no máximo um retalho artificial pode ser gerado em cada padrão de corte. Porém, se o padrão de corte já possui pelo menos um retalho natural, então nenhum retalho artificial pode ser gerado, mesmo que as sobras resultantes não sejam aceitáveis, ou seja, com a imposição destas restrições, é possível que nem toda sobra indesejável seja eliminada.

Esta classificação para o retalho é considerada apenas para os objetos padronizados e não padronizados grandes do estoque (objetos $k, k = 1 \dots, P$). Para os objetos não padronizados pequenos $k, k = P + 1 \dots, K$, conforme a restrição (3.4), as sobras nos padrões de corte são classificada apenas como *retalho natural* ou *perda aceitável*, isto é, os padrões gerados para estes objetos são aceitos sem que qualquer tipo de alteração seja realizada.

Para evitar perdas elevadas em um padrão que não possui retalhos naturais, selecionamos a perda indesejável com maior área e alteramos o nó final que contém a perda selecionada. A Figura 5.7 a seguir ilustra as posições exatas em que as sobras são geradas em um nó final e a estratégia utilizada para transformar uma perda indesejável (perda nos objetos $k = 1, \dots, P$) em retalho artificial. Observe que o nó final é gerado por um padrão de corte homogêneo, ou seja, somente itens de um mesmo tipo compõem o nó final.

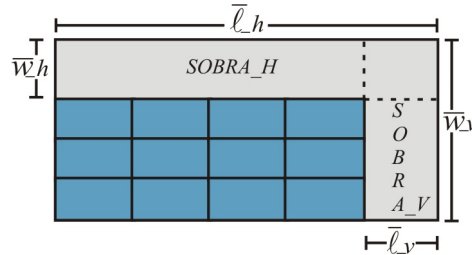


Figura 5.7: Localização das sobras em um nó final.

Na Figura 5.7, $(\bar{\ell}_h, \bar{w}_h)$ são as dimensões da *SOBRA_H* e $(\bar{\ell}_v, \bar{w}_v)$ são dimensões da *SOBRA_V*. Desta forma, se a perda indesejável estiver localizada apenas na posição *SOBRA_H*, camadas de itens da posição horizontal são removidas (uma camada de cada vez) até que um retalho artificial seja gerado, ou seja, até que a condição (3.1) seja satisfeita, caso contrário, camadas de itens da posição vertical serão removidas. Entretanto, é possível que perdas indesejáveis estejam localizadas simultaneamente em

SOBRA_H e *SOBRA_V*. Neste caso, camadas de itens devem ser retiradas (uma camada de cada vez) da posição (horizontal ou vertical) em que temos a perda indesejável com maior área. Esta estratégia de seleção da perda indesejável foi adotada para evitar que muitos itens fossem retirados do nó final para obtermos um retalho artificial.

A seguir, ilustramos na Figura 5.8 o padrão de corte gerado pela abordagem grafo E/OU da Figura 5.6 e o padrão de corte resultante após as alterações realizadas para evitar perdas indesejáveis.

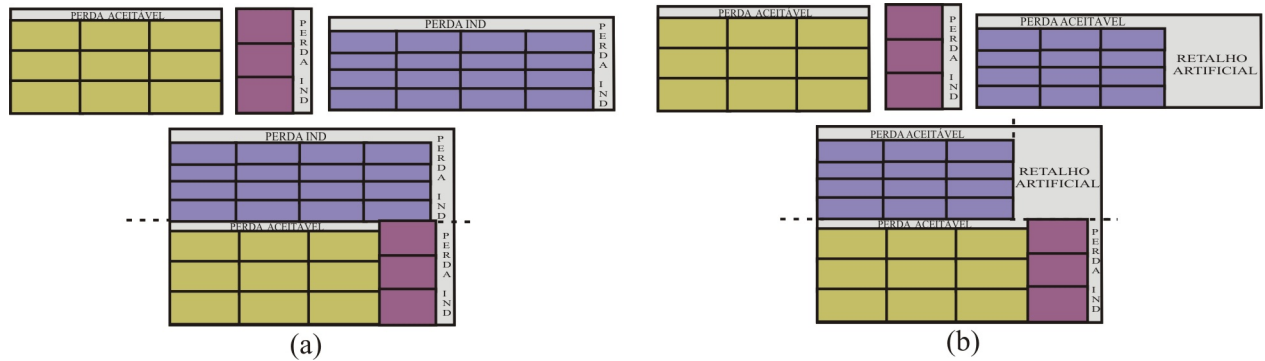


Figura 5.8: (a) padrão de corte gerado pelo grafo E/OU; (b) padrão de corte com alterações.

Pela Figura (5.8 - b) observamos que mesmo após as alterações realizadas, o padrão de corte ficou com uma perda indesejável pois, para eliminá-la, teríamos que gerar outro retalho artificial e, pelos critérios estabelecidos anteriormente, é inaceitável. Entretanto, como eliminamos a perda indesejável com maior área, a perda total do padrão de corte diminuiu.

Observe que antes das alterações realizadas, o padrão de corte era composto por 3 perdas indesejáveis e após as alterações ficou com apenas uma perda indesejável. A eliminação de uma perda indesejável, é consequência da remoção da camada vertical de itens para gerar o retalho artificial. Com a remoção realizada, a perda diminuiu e como os critérios (3.2) ou (3.3) foram satisfeitos, esta perda foi classificada como aceitável. Os itens retirados do padrão de corte retornam a demanda e são cortados utilizando outra placa do estoque.

Este processo de retirada de itens pode ter como consequência um padrão nulo. Neste caso, o padrão inicial, sem as alterações, deve ser aceito, apesar das sobras indesejáveis.

A seguir, apresentamos os procedimentos desenvolvidos para análise e classificação das sobras e a estrutura do grafo E/OU com as alterações realizadas para o aproveitamento de sobras (Grafo E/OU_A).

Seja $\{\bar{1}, \dots, \bar{k}\}$ o conjunto de índices dos objetos padronizados em estoque, $\{\bar{k} + 1, \dots, P\}$ o conjunto de índices dos objetos não padronizados grandes (isto é, objetos com área superior a 50% da área do menor objeto padronizado em estoque) e $\{P + 1, \dots, K\}$ o conjunto de índices dos objetos não padronizados pequenos.

Nos procedimentos desenvolvidos, $(\bar{\ell}_{v_k}, \bar{w}_{v_k})$ e $(\bar{\ell}_{h_k}, \bar{w}_{h_k})$ representam as dimensões de *SOBRA_V_k* e *SOBRA_H_k*, respectivamente (Figura 5.7), *prof* indica a profundidade no grafo E/OU da sobra em análise e $(\bar{\ell}_s, \bar{w}_s)$ ou $(\bar{w}_s, \bar{\ell}_s)$ representam as dimensões estabelecidas a priori para que uma sobra seja classificada como *perda aceitável* ou *retalho natural* nos objetos não padronizados pequenos (objetos $k = P + 1, \dots, K$).

O procedimento **Perda Aceitável** verifica se cada sobra presente nos nós finais dos padrões de corte é uma *perda aceitável*. Neste procedimento, $\rho, 0 < \rho < 1$ (Seção 3.3) é utilizado para evitar que perdas com área grande ($A_s > \xi L_k W_k$, (Seção 3.3)), porém, com uma das dimensões muito pequena e a outra muito grande sejam classificadas como indesejáveis e ψ_k representa a porcentagem de perda mínima admissível para cada placa k em estoque $k = 1, \dots, P$.

Procedimento Perda Aceitável

Se $\bar{\ell}_{v_k} \cdot \bar{w}_{v_k} > \bar{\ell}_{h_k} \cdot \bar{w}_{h_k}$ então

$$\bar{\ell}_k = \bar{\ell}_{v_k} \text{ e } \bar{w}_k = \bar{w}_{v_k};$$

Senão

$$\bar{\ell}_k = \bar{\ell}_{h_k} \text{ e } \bar{w}_k = \bar{w}_{h_k};$$

Se $k \leq P$ então

Início

Se $prof \leq 2$ então

Se $(\bar{\ell}_k \cdot \bar{w}_k < \frac{\psi_k L_k W_k}{2^{prof}})$ **ou** $(\bar{\ell}_k \leq \rho L_k \text{ ou } \bar{w}_k \leq \rho W_k)$ então

$$SOBRA_V_k = SOBRA_H_k = \textit{perda aceitável};$$

Senão

Se $(\bar{\ell}_k \cdot \bar{w}_k < \frac{\psi_k L_k W_k}{4})$ **ou** $(\bar{\ell}_k \leq \rho L_k \text{ ou } \bar{w}_k \leq \rho W_k)$ então

$SOBRA_V_k = SOBRA_H_k = \text{perda aceitável};$

Fim

Senão

Início

Se $(\bar{\ell}_k < \bar{\ell}_s \text{ e } \bar{w}_k < \bar{w}_s)$ **ou** $(\bar{\ell}_k < \bar{w}_s \text{ e } \bar{\ell}_k < \bar{w}_s)$ então

$SOBRA_V_k = SOBRA_V_k = \text{perda aceitável};$

Fim

No procedimento Perda Aceitável, sempre realizamos a análise para as sobras com maiores dimensões. Assim, desde que uma sobra ($SOBRA_V_k$ ou $SOBRA_H_k$) satisfaz um dos critérios que a classificam como perda aceitável, a outra sobra (que é menor) também é classificada como perda aceitável.

Para verificar se as sobras geradas em um padrão de corte é um retalho natural, considere ϕ , em que $0 < \phi < 1$ (Seção 3.3). O procedimento **Retalho Natural** é apresentado a seguir.

Procedimento Retalho Natural

Se $k \leq P$ então

Início

Se $\bar{\ell}_{.v_k} \cdot \bar{w}_{.v_k} \geq \bar{\ell}_{.h_k} \cdot \bar{w}_{.h_k}$ então

Início

Se $\bar{\ell}_{.v_k} \geq \phi L_k$ e $\bar{w}_{.v_k} \geq \phi W_k$ então

$SOBRA_V_k = \text{retalho natural};$

Se $(\bar{\ell}_{.h_k} - \bar{\ell}_{.v_k}) \geq \phi L_k$ e $\bar{w}_{.h_k} \geq \phi W_k$ então

$SOBRA_H_k = SOBRA_H_k - (\bar{\ell}_{.v_k} \cdot \bar{w}_{.h_k});$

$SOBRA_H_k = \text{retalho natural};$

Fim

Senão

Início

Se $\bar{\ell}_{.h_k} \geq \phi L_k$ e $\bar{w}_{.h_k} \geq \phi W_k$ então

$SOBRA_H_k = \text{retalho natural};$

Se $\bar{\ell}_{.v_k} \geq \phi L_k$ e $(\bar{w}_{.v_k} - \bar{w}_{.h_k}) \geq \phi W_k$ então

$SOBRA_V_k = SOBRA_V_k - (\bar{\ell}_{.v_k} \cdot \bar{w}_{.h_k});$

$SOBRA.V_k = \text{retalho natural};$

Fim

Senão

Se $\bar{\ell}.v_k \cdot \bar{w}.v_k \geq \bar{\ell}.h_k \cdot \bar{w}.h_k$ então

Início

Se $(\bar{\ell}.v_k \geq \bar{\ell}_s \text{ e } \bar{w}.v_k \geq \bar{w}_s)$ **ou** $(\bar{\ell}.v_k \geq \bar{w}_s \text{ e } \bar{w}.v_k \geq \bar{\ell}_s)$ então

$SOBRA.V_k = \text{retalho natural};$

Se $(\bar{\ell}.h_k - \bar{\ell}.v_k) \geq \bar{\ell}_s \text{ e } \bar{w}.h_k \geq \bar{w}_s)$ **ou** $(\bar{\ell}.h_k - \bar{\ell}.v_k) \geq \bar{w}_s \text{ e } \bar{w}.v_k \geq \bar{\ell}_s)$ então

$SOBRA.H_k = SOBRA.H_k - (\bar{\ell}.v_k \cdot \bar{w}.h_k);$

$SOBRA.H_k = \text{retalho natural};$

Senão $SOBRA.H_k = \text{perda aceitável};$

Fim

Senão

Início

Se $(\bar{\ell}.h_k \geq \bar{\ell}_s \text{ e } \bar{w}.h_k \geq \bar{w}_s)$ **ou** $(\bar{\ell}.h_k \geq \bar{w}_s \text{ e } \bar{w}.h_k \geq \bar{\ell}_s)$ então

$SOBRA.H_k = \text{retalho natural};$

Se $(\bar{\ell}.v_k \geq \bar{\ell}_s \text{ e } (\bar{w}.v_k - \bar{w}.h_k) \geq \bar{w}_s)$ **ou** $(\bar{\ell}.h_k \geq \bar{w}_s \text{ e } (\bar{w}.v_k - \bar{w}.h_k) \geq \bar{\ell}_s)$ então

$SOBRA.V_k = SOBRA.V_k - (\bar{\ell}.v_k \cdot \bar{w}.h_k);$

$SOBRA.V_k = \text{retalho natural};$

Senão $SOBRA.V_k = \text{perda aceitável};$

Fim

Fim

Após analisar as sobras presentes nos nós finais, um padrão é determinado caso estas sobras sejam aceitáveis (*perda* ou *retalho*). Observe que nos retalhos pequenos ($k = P + 1, \dots, K$), toda sobra é classificada como *perda aceitável* ou *retalho natural*. Entretanto, para os demais objetos ($k = 1, \dots, P$), uma sobra pode não ser classificada como aceitável e, portanto, é uma *sobra indesejável*. Neste caso, desde que o padrão não tenha nenhum *retalho natural*, alterações devem ser realizadas no nó final que compõe este padrão conforme o procedimento **Retalho Artificial** a seguir.

Procedimento Retalho Artificial

Se $\bar{\ell}.v_k \cdot \bar{w}.v_k > \bar{\ell}.h_k \cdot \bar{w}.h_k$ então

Início

Enquanto $\bar{\ell}_k < \phi L_k$ faça

 Remover camadas verticais de itens do nó final;

$SOBRA_V_k = \text{Retalho artificial}$;

 Recalcular $SOBRA_H_k$;

Fim

Senão

Início

Enquanto $\bar{w}_k < W_k$ faça

 Remover camadas horizontais de itens do nó final;

$SOBRA_H_k = \text{Retalho artificial}$;

 Recalcular $SOBRA_V_k$;

Fim

Estes procedimentos descritos são utilizados como rotinas no grafo E/OU que considera o aproveitamento de sobras. A seguir, apresentamos os principais passos deste algoritmo.

Algoritmo Grafo E/OU_A

k : índice de uma placa em estoque;

Passo 1: {Início}

 Resolver o problema de corte utilizando a abordagem grafo E/OU e obter um padrão de corte para a placa k em estoque;

Passo 2: {Análise dos nós finais da placa k }

 Seja $FINAL$ o conjunto de nós finais da placa k e $|FINAL|$ sua cardinalidade;

 Retalho = 0;

 Enquanto $|FINAL| > 0$ faça

 Início

 Escolher e retirar um nó final de $FINAL$;

$|FINAL| = |FINAL| - 1$;

 Se $SOBRA_V_k = SOBRA_H_k = 0$ então

 Nó final aceito;

 Senão

Início

$SOBRA.V_k = SOBRA.H_k = perda\ indesejável;$

Se $SOBRA.V_k = 0$ então $SOBRA.V_k = perda\ aceitável;$

Se $SOBRA.H_k = 0$ então $SOBRA.H_k = perda\ aceitável;$

Procedimento Perda Aceitável;

Se $(SOBRA.V_k = perda\ indesejável)$ ou $(SOBRA.H_k = perda\ indesejável)$ então

Procedimento Retalho Natural;

Se $(SOBRA.V_k = retalho\ natural)$ ou $(SOBRA.H_k = retalho\ natural)$ então

$Retalho = Retalho + 1;$

Se $(k \leq P)$ e $(Retalho = 0)$ e $(SOBRA.V_k = perda\ indesejável)$ ou $(SOBRA.H_k = perda\ indesejável)$ então

Procedimento Retalho Artificial;

Fim

Fim

Passo 3: {Viabilidade da solução}

Se o padrão de corte k gerado no Passo 2 for nulo então

armazenar o padrão de corte gerado no Passo 1;

Senão

armazenar o padrão de corte gerado no Passo 2;

O algoritmo Grafo E/OU_A utiliza o grafo E/OU para gerar um padrão de corte para cada tipo de placa em estoque e, em seguida, realiza algumas modificações para considerar o aproveitamento de sobras. Com as alterações realizadas este algoritmo pode ser utilizado em procedimentos heurísticos que resolvem o problema de corte de estoque com sobras aproveitáveis, quando uma demanda deve ser atendida.

A seguir apresentamos as heurísticas Gulosa_A^{2D} e RAG_A^{2D} desenvolvidas para resolver o problema de corte de estoque bidimensional com sobras aproveitáveis.

5.3 Heurística Gulosa_A^{2D}

A heurística Gulosa_A^{2D} é uma heurística de repetição exaustiva que possui a mesma estrutura da heurística Gulosa (Seção 4.1), entretanto, para considerar o aproveitamento

de sobras, os padrões de corte são definidos pelo algoritmo Grafo E/OU_A.

Além de considerar o aproveitamento de sobras, a heurística Gulosa_A^{2D} também prioriza o uso dos objetos não padronizados disponíveis no estoque. Para isto, disponibilizamos inicialmente para o processo de corte os objetos não padronizados pequenos (objetos $k = P + 1, \dots, K$). Quando todos estes objetos tiverem sido cortados, disponibilizamos os objetos padronizados e não padronizados grandes (objetos $k = 1, \dots, P$) para o corte. Observe que, a princípio, é possível atender toda a demanda apenas com o corte dos objetos não padronizados pequenos.

A seguir, apresentamos os principais passos da heurística Gulosa_A^{2D}. Neste algoritmo, denotamos por $e_k, k = 1, \dots, K$ a quantidade de objetos do tipo k em estoque.

Heurística Gulosa_A^{2D}

Passo 1: Aplique o algoritmo Grafo E/OU_A para $k = P + 1, \dots, K$;

Passo 2: Escolha um dos padrões de corte gerados no Passo 1 (menor perda e menor quantidade de retalhos);

Passo 3: Use o padrão de corte escolhido no Passo 2 o máximo possível, sem exceder a demanda dos itens e a disponibilidade do objeto associado;

Passo 4: Atualize a demanda dos itens e o estoque dos objetos. Se a demanda dos itens for atendida, PARE;

Passo 5: Se $e_k = 0, k = P + 1, \dots, K$, vá para o Passo 6, senão, vá para o Passo 1;

Passo 6: Aplique Grafo E/OU_A para $k = 1, \dots, P$ e vá para o Passo 2.

Devido a estratégia gulosa, bons padrões de corte são usados inicialmente (perdas pequenas e sem retalhos), enquanto que os últimos padrões de corte, compostos por itens maiores (mais difíceis de serem combinados), tendem a apresentar um número maior de retalhos. Entretanto, com a prioridade de corte atribuída aos objetos não padronizados pequenos em relação aos demais objetos, o estoque destes objetos tende a ser reduzido.

Além da heurística Gulosa_A^{2D} também desenvolvemos um procedimento heurístico residual que considera o aproveitamento de sobras para problemas de corte bidimensionais. Este procedimento, que também prioriza o corte dos objetos não padronizados, é descrito a seguir.

5.4 Heurística RAG_A^{2D}

Considere o modelo matemático (Seção 4.2) proposto por Gilmore e Gomory (1963):

$$\text{minimizar } f(x) = \sum_{k=1}^K \sum_{j=1}^{N_k} c_{jk} x_{jk} \quad (5.6)$$

sujeito a:

$$\sum_{k=1}^K \sum_{j=1}^{N_k} \alpha_{ijk} x_{jk} = d_i, \quad i = 1, \dots, m \quad (5.7)$$

$$\sum_{k=1}^{N_k} x_{jk} \leq e_k, \quad k = 1, \dots, K \quad (5.8)$$

$$x_{jk} \geq 0 \text{ e inteiro, } j = 1, \dots, N_k, \quad k = 1, \dots, K \quad (5.9)$$

Para resolver o problema de corte bidimensional, este mesmo modelo pode ser utilizado para obtermos uma solução ótima relaxada, entretanto, consideramos:

- c_{jk} é a sobra de material no padrão de corte j para a placa do tipo k em estoque dada por $c_{jk} = (L_k W_k) - \sum_{i=1}^m (\ell_i w_i) \alpha_{ijk}$, $j = 1, \dots, N_k$, $k = 1, \dots, K$;
- x_{jk} é a variável de decisão que representa o número de placas do tipo k cortadas de acordo com o padrão j , $j = 1, \dots, N_k$, $k = 1, \dots, K$;
- $\alpha_{jk} = (\alpha_{1jk}, \alpha_{2jk}, \dots, \alpha_{mjk})$ é um vetor associado ao padrão de corte j em que α_{ijk} é o número de itens do tipo i no padrão de corte j para o objeto do tipo k , $i = 1, \dots, m$, $j = 1, \dots, N_k$, $k = 1, \dots, K$.

A estrutura da heurística RAG_A^{2D} é a mesma da heurística RAG_A^P (Seção 4.2.4), apenas alterando-se a geração de colunas, que são associadas a padrões de corte bidimensionais guilhotinados, obtidos pela abordagem grafo E/OU.

Para a heurística RAG_A^{2D} , estabelecemos limitantes para que uma sobra seja classificada como *perda aceitável* nos objetos padronizados, não padronizados grandes e não padronizados pequenos. Os parâmetros utilizados para classificar as perdas são calculados considerando a solução do problema de corte bidimensional obtida utilizando uma das versões das heurísticas RAG sem as alterações para o aproveitamento de sobras. Desta forma, definimos:

$$\theta = \frac{\sum_{k=1}^P \sum_{j=1}^{N_k} \left(L_k W_k - \sum_{i=1}^m \ell_i w_i \alpha_{ijk} - ARN_{jk} \right) x_{jk}}{\sum_{k=1}^P \sum_{j=1}^{N_k} L_k W_k x_{jk}}$$

$$\beta = \frac{\sum_{k=P+1}^K \sum_{j=1}^{N_k} \left(L_k W_k - \sum_{i=1}^m \ell_i w_i \alpha_{ijk} - ARN_{jk} \right) x_{jk}}{\sum_{k=P+1}^K \sum_{j=1}^{N_k} L_k W_k x_{jk}}$$

em que:

- ARN_{jk} : área do *retalho natural* no padrão de corte j para a placa $k, k = 1, \dots, K$;
- θ : fração que define a área máxima para que uma *sobra* seja considerada *perda pequena* (aceitável) em um objeto padronizado e não padronizado grande, ou seja, $\theta L_k W_k$ é a área máxima para que uma sobra seja considerada pequena em um objeto com dimensões $(L_k \times W_k), k = 1, \dots, P$ (P é o número de tipos de objetos padronizados e não padronizados grandes disponíveis no estoque);
- β : fração que define a área máxima para que uma *sobra* seja considerada *perda pequena* em um objeto não padronizado pequeno, ou seja, $\beta L_k W_k$ é a área máxima para que uma sobra seja considerada pequena em um objeto com dimensões $(L_k \times W_k), k = P + 1, \dots, K$ (os objetos do tipo $P + 1, \dots, K$ são retalhos pequenos e K é o número total de tipos de objetos disponíveis no estoque);

Para priorizar o corte dos objetos não padronizados do estoque, na função objetivo do modelo (5.6)-(5.9) consideramos c_{jk} o coeficiente de custo associado a cada placa do tipo k em estoque, $k = 1, \dots, K$. Com este coeficiente, custos mais baixos são associados aos objetos não padronizados (objetos $k = \bar{k} + 1, \dots, K$) como segue,

$$c_{jk} = \begin{cases} c(L_k \times W_k), & k = 1, \dots, \bar{k} \\ \mu(c(L_k \times W_k)), & k = \bar{k} + 1, \dots, P \\ \gamma(c(L_k \times W_k)), & k = P + 1, \dots, K \end{cases} \quad (5.10)$$

em que c é um custo por unidade de área e μ e γ são descontos que incentivam o uso de objetos não padronizados grandes e pequenos, respectivamente, $0 < \gamma \leq \mu < 1$.

Além de atribuímos custos diferenciados para os objetos em estoque, na heurística RAG_A^{2D} o problema residual formado pelos padrões de corte rejeitados é resolvido pela heurística Gulosa $_A^{2D}$ que também prioriza o corte de objetos não padronizados.

Utilizando os parâmetros θ e β definidos para classificar as sobras como *perda pequena* e λ_{jk_j} a sobra (retalhos não são considerados) gerada no padrão de corte j para a placa k_j , $j = 1, \dots, T$, em que T é a quantidade de padrões de corte com freqüências não nulas, apresentamos os principais passos da heurística RAG_A^{2D} .

Algoritmo RAG_A^{2D}

Passo 1: {Início}

Determine uma solução inteira aproximada para o problema (5.6)-(5.9), com os coeficientes de custo conforme (5.10), utilizando uma das versões do procedimento de arredondamento guloso RAG (Seção 4.2.2) e obtenha T padrões de corte com freqüências positivas, em que o padrão j está associado ao objeto k_j , $j = 1, \dots, T$;

Passo 2: {Análise das perdas geradas pelos padrões}

Para cada padrão de corte j gerado no Passo 1, se a sobra da placa for aceitável, isto é, $\lambda_{jk_j} \leq \theta L_k W_k$, $k_j \in \{1, 2, \dots, P\}$ ou $\lambda_{jk_j} \leq \beta L_k W_k$, $k_j \in \{P + 1, \dots, K\}$, então o padrão de corte j para o objeto k_j é aceito e armazenado com a freqüência obtida no Passo 1;

Senão, o padrão de corte j é rejeitado (isto é, sua freqüência é fixada em zero). A demanda dos itens no padrão j e estoque dos objetos tipo k_j são atualizados;

Passo 3: {Problema Residual}

Se não houver padrão rejeitado no Passo 2, PARE. Senão, aplique o procedimento Guloso $_A^{2D}$ para resolver o problema residual formado pelos itens dos padrões de corte rejeitados no Passo 2 e objetos remanescentes do estoque.

5.5 Exemplo

Neste exemplo, ilustramos os padrões de corte gerados pelas heurísticas Gulosa $_A^{2D}$ e RAG_A^{2D} utilizando a interface gráfica desenvolvida.

Suponha que temos um estoque composto por $K = 4$ tipos de objetos, sendo $(L_1 \times W_1)$ o objeto padronizado e $(L_k \times W_k), k = 2, 3, 4$, objetos não padronizados pequenos. As dimensões destes objetos, assim como suas disponibilidades $e_k, k = 1, \dots, 4$ estão na Tabela 5.1.

Tabela 5.1: Dados dos objetos em estoque

| Objetos | Dimensões | estoque |
|---------|--------------------|---------|
| 1 | (100×120) | 20 |
| 2 | (56×70) | 2 |
| 3 | (28×48) | 1 |
| 4 | (39×45) | 2 |

Para este exemplo, definimos os parâmetros utilizados pelo algoritmo Grafo E/OU_A para classificar as sobras geradas: $\psi_k = 0,05, k = 1, \dots, P$ (o valor 0,05 é utilizado quando o valor de ψ_k obtido durante o processo de corte é nulo), $\rho = 0,1, \phi = 0,25$ e $(\bar{\ell}_s, \bar{w}_s) = (10, 10)$. Para a heurística RAG_A^{2D} como temos apenas objetos não padronizados pequenos em estoque (objetos $k = P + 1, \dots, K$) consideramos apenas o coeficiente de custo $\gamma = 0,90$.

As demandas a serem atendidas são compostas por $m = 5$ tipos de itens cujos dados são apresentados na Tabela 5.2.

Tabela 5.2: Dados dos itens a serem produzidos

| Itens | Dimensões | demanda |
|-------|------------------|---------|
| 1 | (15×42) | 9 |
| 2 | (22×13) | 12 |
| 3 | (25×28) | 10 |
| 4 | (32×10) | 25 |
| 5 | (18×12) | 17 |

As Figuras 5.9 - 5.20 a seguir ilustram os padrões de corte gerados.

Heurística Gulosa^{2D}_A:

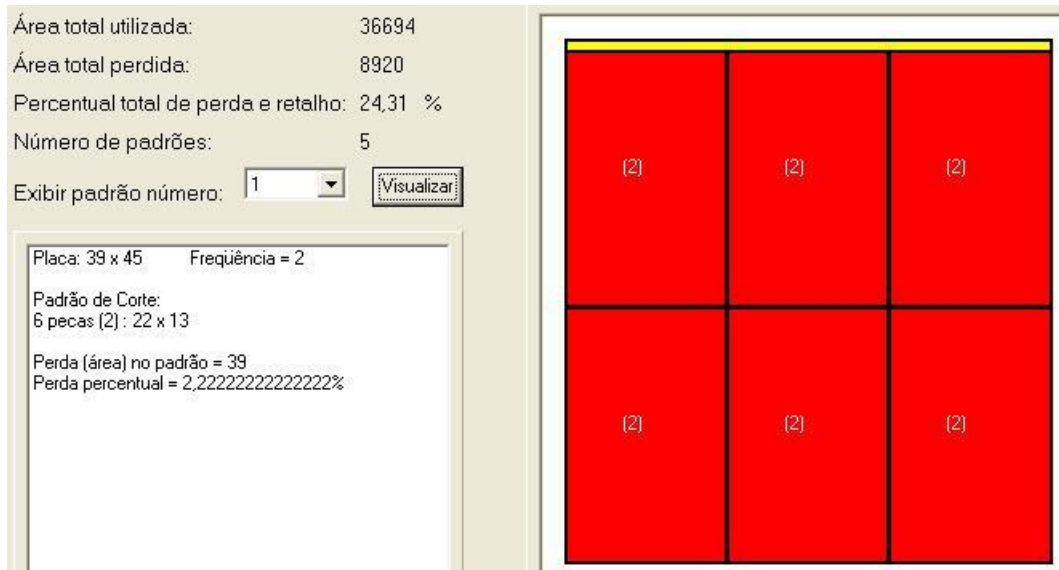


Figura 5.9: Gulosa^{2D}_A: primeiro padrão de corte.

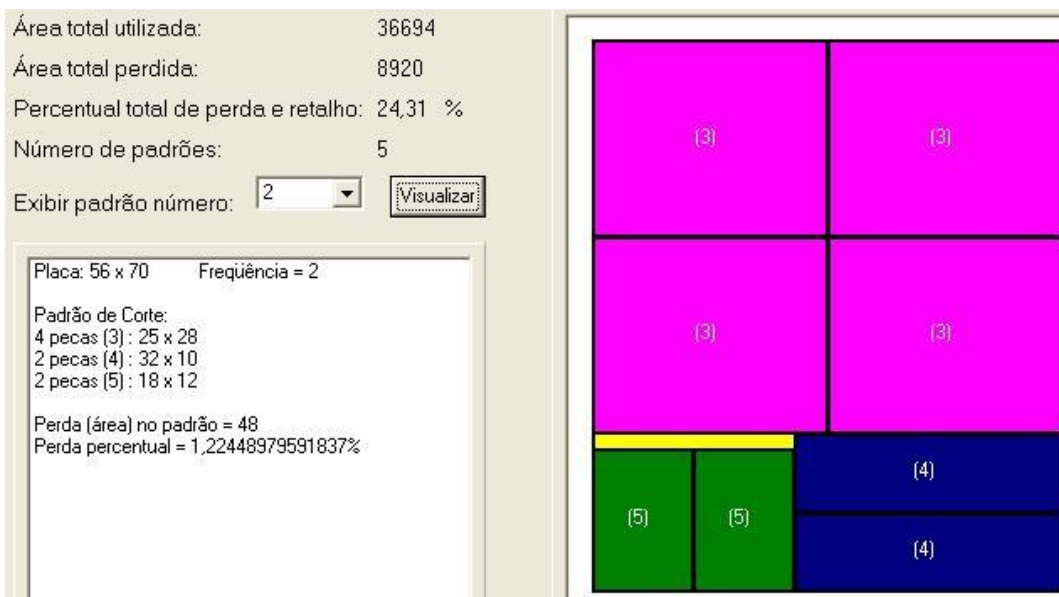


Figura 5.10: Gulosa^{2D}_A: segundo padrão de corte.

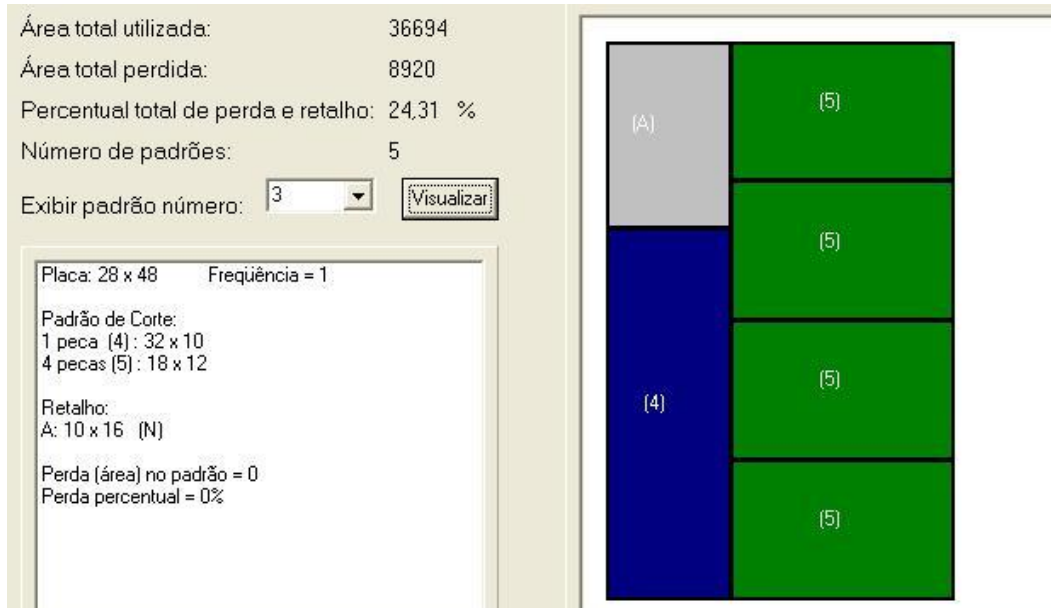


Figura 5.11: Gulosa_A^{2D}: terceiro padrão de corte.

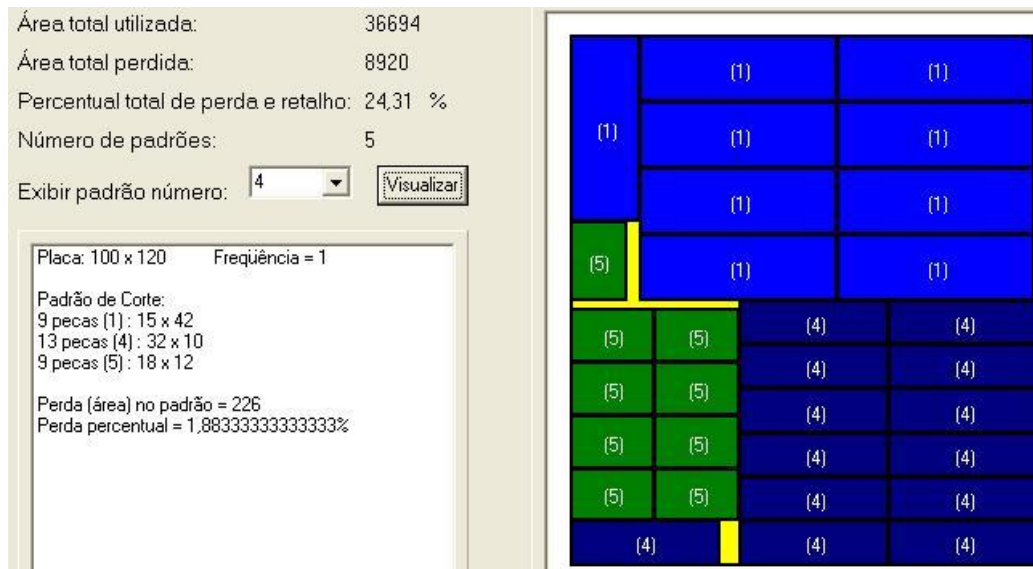


Figura 5.12: Gulosa_A^{2D}: quarto padrão de corte.

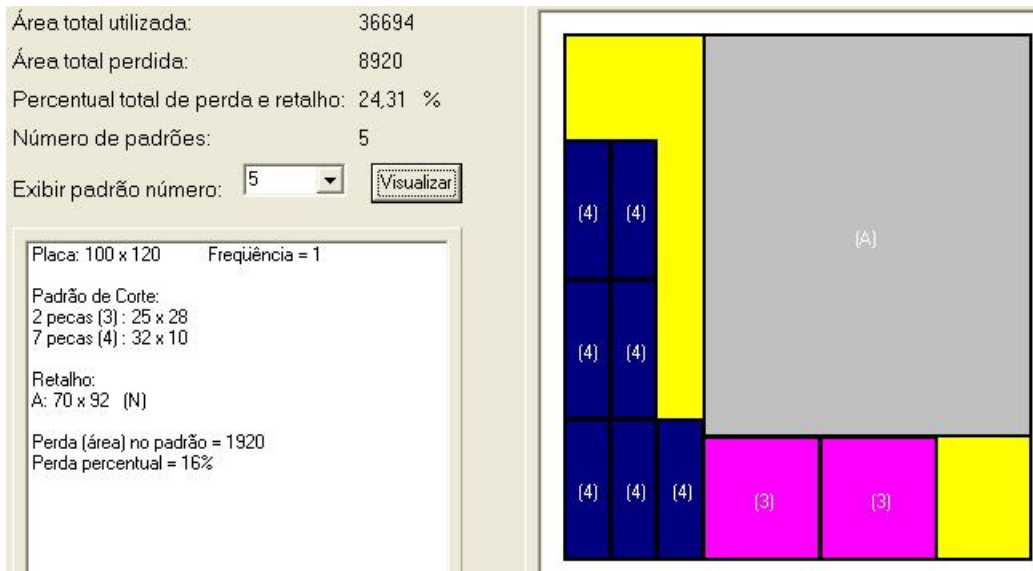


Figura 5.13: $Gulosa_A^{2D}$: quinto padrão de corte.

Heurística RAG_A^{2D} :

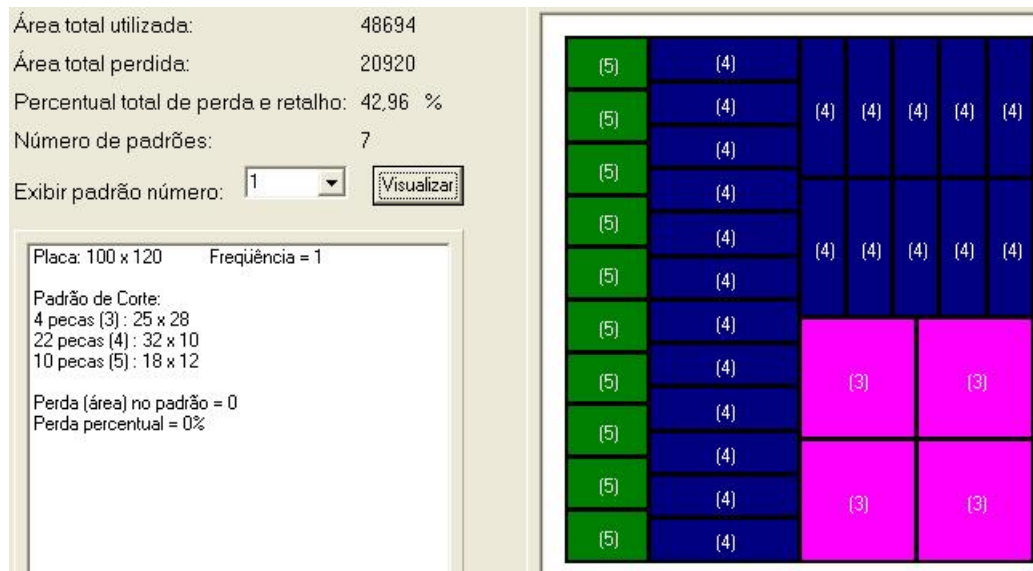


Figura 5.14: RAG_A^{2D} : primeiro padrão de corte.

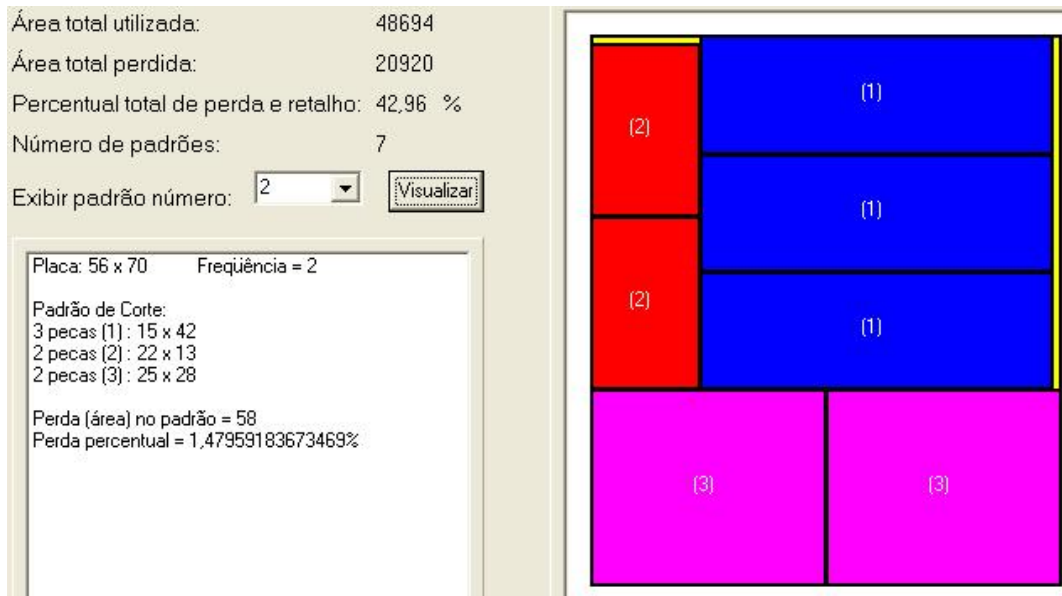


Figura 5.15: RAG_A^{2D} : segundo padrão de corte.

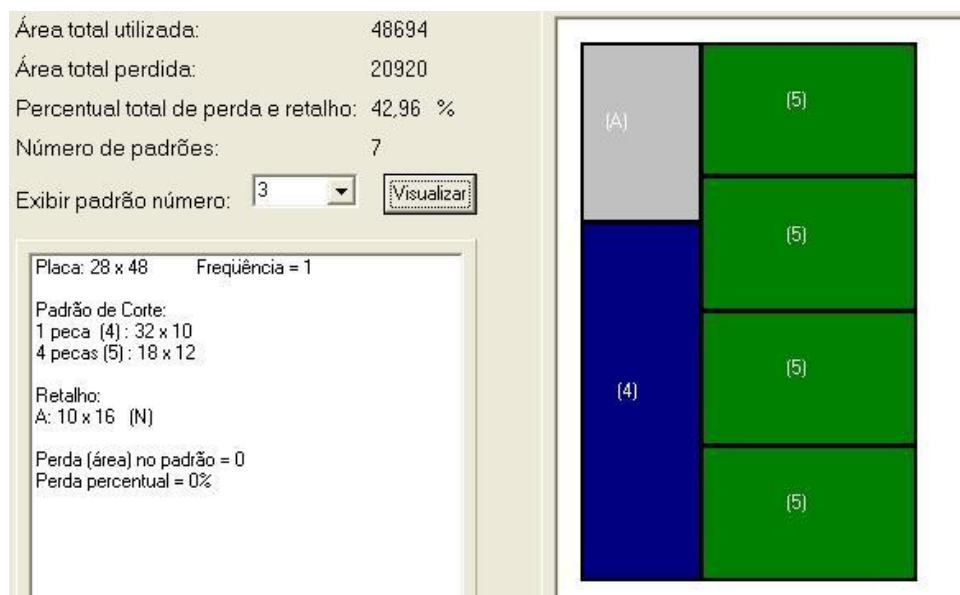


Figura 5.16: RAG_A^{2D} : terceiro padrão de corte.

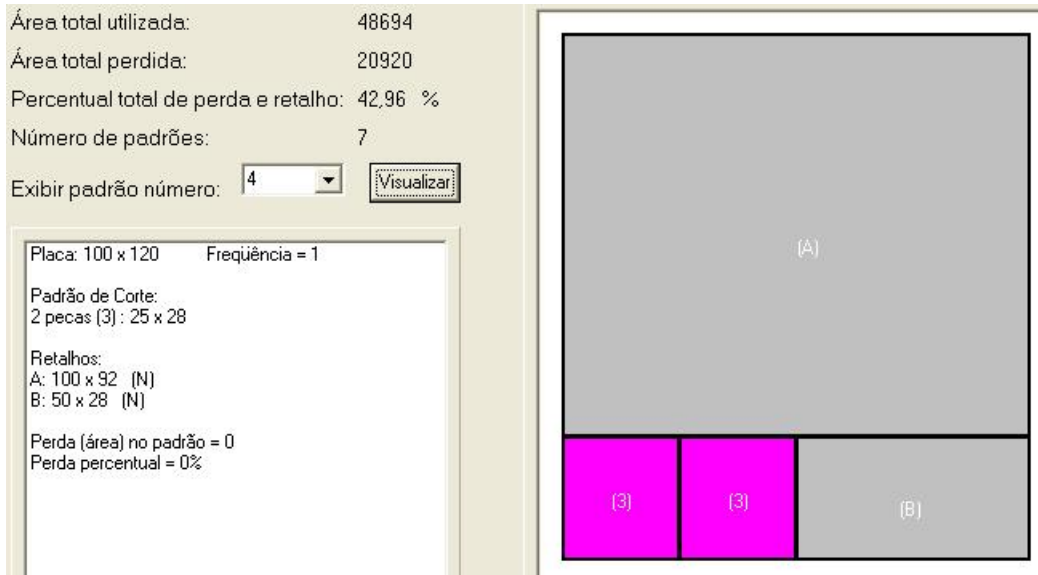


Figura 5.17: RAG_A^{2D} : quarto padrão de corte.

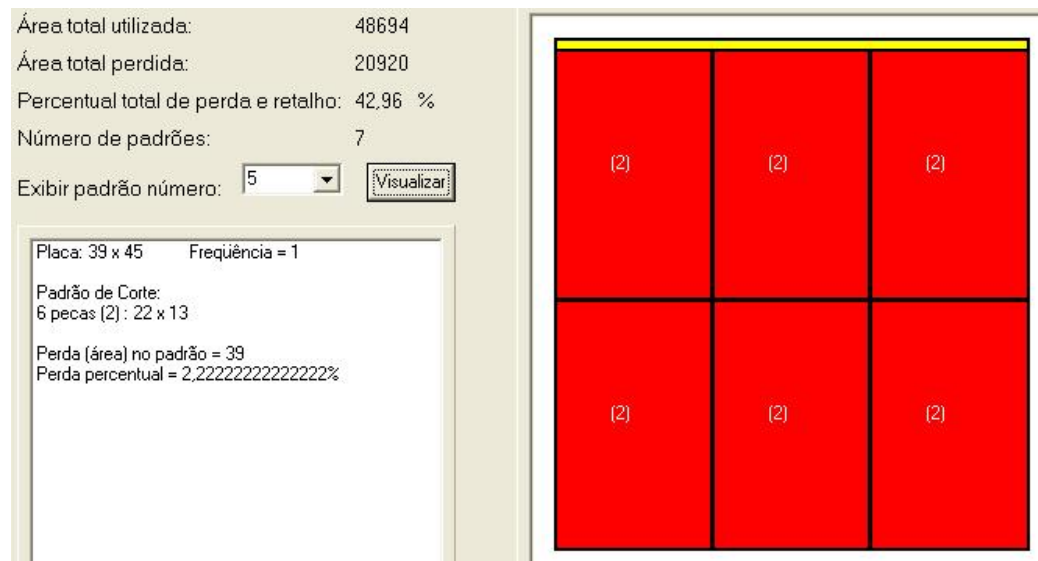


Figura 5.18: RAG_A^{2D} : quinto padrão de corte.

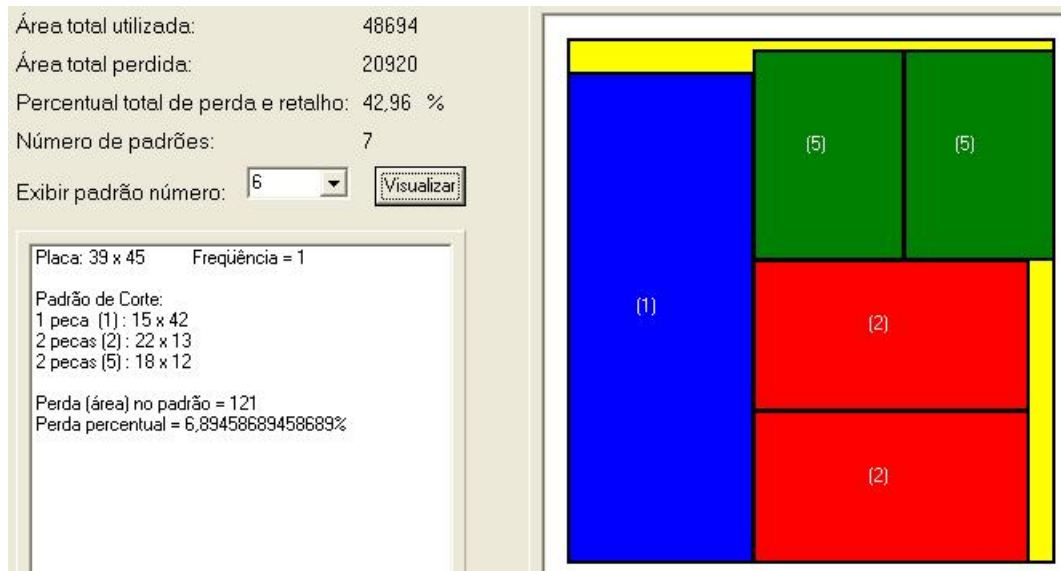


Figura 5.19: RAG_A^{2D} : sexto padrão de corte.

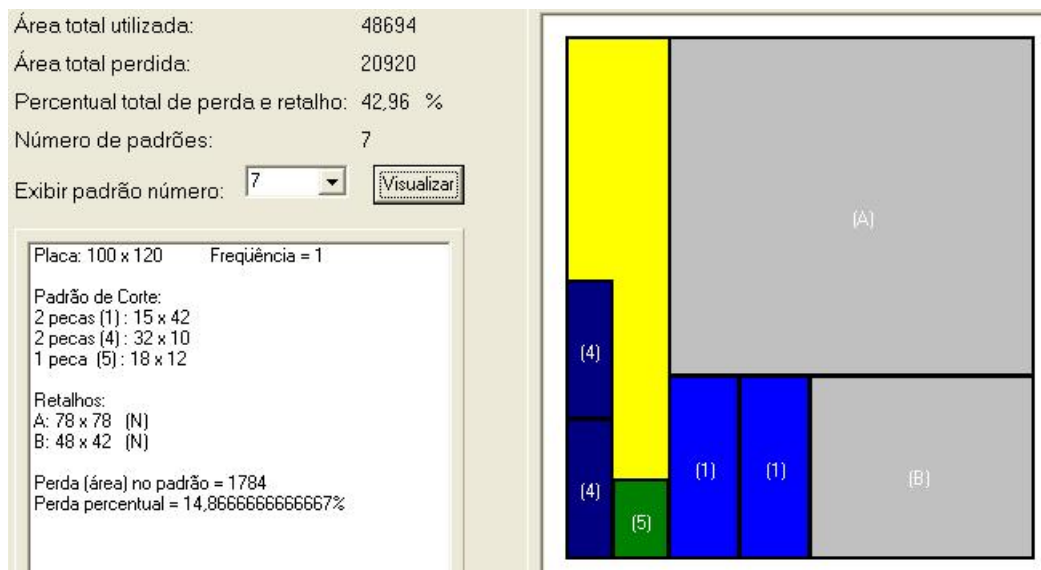


Figura 5.20: RAG_A^{2D} : sétimo padrão de corte.

Utilizando a interface gráfica, temos algumas informações de interesse para analisar as soluções obtidas para o problema de corte com sobras aproveitáveis. Além de informações gerais sobre a solução (perda total, área total utilizada, porcentagem total de perda e retalho e número de padrões de corte), é possível visualizar outras características

referentes a cada padrão de corte.

Com estas informações, observamos que a perda gerada pela heurística RAG_A^{2D} (4,38%) é inferior a perda da heurística $Gulosa_A^{2D}$ (6,68%). Outra característica importante que podemos observar é que as heurísticas utilizaram todos os objetos não padronizados do estoque, porém, a heurística $Gulosa_A^{2D}$ gerou 2 novos *retalhos naturais*, enquanto que a heurística RAG_A^{2D} gerou 5.

Em nossa análise, a solução da heurística RAG_A^{2D} pode ser mais interessante, pois, apresenta a menor perda. Entretanto, para uma empresa, esta solução pode não ser atraente, desde que a quantidade de retalhos que deve ser manipulada e estocada ao final do processo de corte é superior a quantidade da heurística $Gulosa_A^{2D}$.

No Capítulo 6 a seguir, apresentamos os testes computacionais realizados utilizando todos os procedimentos heurísticos desenvolvidos (unidimensional e bidimensional) para resolver o problema de corte com sobras aproveitáveis. Os experimentos foram realizados considerando problemas da literatura, problemas reais e problemas gerados aleatoriamente.

Capítulo 6

Experimentos computacionais

Neste capítulo apresentamos os testes computacionais realizados para analisar o desempenho dos procedimentos heurísticos (Capítulos 4 e 5) que consideram o aproveitamento de sobras para problemas de corte unidimensionais e bidimensionais. Os resultados computacionais obtidos são apresentados separadamente considerando os três problemas descritos no Capítulo 3.

Todos os algoritmos desenvolvidos foram implementados na linguagem de programação DELPHI 6 e os testes foram executados em um microcomputador Pentium IV (3 GHz, com 2 GB de memória RAM).

6.1 Problema de corte unidimensional com sobras aproveitáveis

Nesta seção apresentamos os resultados computacionais obtidos pelos procedimentos heurísticos que não priorizam o corte dos objetos não padronizados (retalhos). Para avaliar estes procedimentos descritos no Capítulos 4, 16 classes de exemplos foram consideradas e, para cada classe, 20 exemplares foram gerados aleatoriamente. Para estas classes geradas aleatoriamente, também apresentamos as soluções obtidas pelo algoritmo COLA, desenvolvido por Gradisar *et al.* (1997). Este algoritmo tem a finalidade de minimizar a perda de material ou então concentrá-las em um objeto cortado de maneira que se tornem retalho. Como limite para a sobra, o algoritmo COLA considera o com-

primento do menor item demandado, entretanto, como comparamos suas soluções com as soluções obtidas pelos procedimentos heurísticos desenvolvidos, consideramos como limite para a sobra, todo pedaço cortado (não demandado) com comprimento superior ou igual à média dos comprimentos dos itens demandados. Ainda para análise do desempenho das heurísticas de aproveitamento de sobras propostas, realizamos testes com os exemplos de Trkman (2005) e com alguns exemplos práticos apresentados em Abuabara e Morabito (2008).

Para classificar as soluções obtidas nos testes computacionais realizados conforme a Definição 3.1 em *ideal*, *aceitável*, ou *indesejável*, utilizamos os valores $\varepsilon_1 = 0,03$ e $\varepsilon_2 = 0,1$ (Seção 3.1), exceto para os exemplos 9 a 11 que apresentam uma demanda muito pequena.

Para análise das soluções conforme Definição 3.1, são necessárias apenas as informações sobre o número de objetos cortados com perdas pequenas, não tão pequenas ou com retalhos. Entretanto, exibimos outros resultados que podem ser úteis para o decisor na escolha de uma solução (os conceitos de *solução ideal*, *aceitável* ou *indesejável* podem ser revisados pelo usuário e incluir outras características). Os dados adicionais que apresentamos são: número de objetos cortados (padronizados e não padronizados), comprimento total perdido (soma das perdas) e comprimento total de novos objetos não padronizados gerados (soma dos retalhos).

Note que no problema de corte de estoque com sobras aproveitáveis, enquanto é desejável que o comprimento total perdido seja pequeno, não necessariamente é desejável que o comprimento total de retalhos seja pequeno.

6.1.1 Resultados de exemplares da literatura

Nesta seção, apresentamos exemplos numéricos de Trkman (2005) e exemplos práticos de Abuabara e Morabito (2008).

Nos exemplares de Trkman (2005) temos vários tipos de objetos em estoque, porém, a disponibilidade é de apenas um objeto para cada tipo, ou seja, $e_k = 1$ para todo k . Esta é uma situação especial em que as variáveis x_{jk} são binárias. Devido a característica especial destes exemplos, consideramos que o estoque é composto apenas por objetos não padronizados e desta forma, fixamos o parâmetro β em 0,005.

Para os exemplos apresentados a seguir, temos as informações do algoritmo CUT. Desta forma, utilizamos os mesmos critérios adotados por Gradisar *et al.* (1997) para definir *perda* e *retalho*, ou seja, toda sobra de material em um padrão de corte com comprimento maior ou igual ao comprimento do menor item demandado é considerada retalho, caso contrário, é considerada perda. Para estes exemplos, comparamos as soluções do algoritmo CUT com as soluções dos procedimentos heurísticos que consideram o aproveitamento de sobras.

Nas tabelas a seguir, classificamos as soluções conforme a Definição 3.1 e utilizamos **ID** para denotar uma *solução ideal*, **AC** como *solução aceitável* e **IND** para denotar uma *solução indesejável*. Além destas notações, também utilizamos (Obj.Cort.) para denotar a quantidade de objetos cortados durante o processo de corte, (Comp.Total) representa o comprimento total cortado dos objetos em estoque, (Perda Total) denota o comprimento total das perdas, (Ret. Total) representa o comprimento total dos retalhos, (PPequena) denota o número de objetos cortados com perda pequena, (PNPequena) denota o número de objetos cortados com perda não tão pequena e, finalmente, (Retalho) representa o número de objetos cortados com retalho.

Exemplo 1: Temos em estoque $K = 20$ tipos de objetos com comprimentos entre 2200 e 6000 cm e disponibilidade de uma unidade para cada tipo de objeto. A demanda é composta por $m = 5$ tipos de itens cujos dados estão na Tabela 6.1.

Tabela 6.1: Dados do Exemplo 1: Itens

| Item | Comprimento (cm) | Demanda |
|------|------------------|---------|
| 1 | 437 | 13 |
| 2 | 516 | 33 |
| 3 | 296 | 37 |
| 4 | 302 | 12 |
| 5 | 345 | 55 |

Para este exemplo, o limite inferior para uma sobra ser classificada como retalho é $\delta = 296$ cm. Na Tabela 6.2 apresentamos os resultados obtidos.

Tabela 6.2: Solução do Exemplo 1

| | <i>CUT</i> | Construtiva | | Residual | | | | |
|-------------|------------|------------------------|---------------------------|------------------------|---------------------------|--------------------------|--------------------------|--------------------------|
| | | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>RAG_A 1</i> | <i>RAG_A 2</i> | <i>RAG_A 3</i> |
| Obj. Cort. | 14 | 17 | 17 | 16 | 16 | 15 | 15 | 16 |
| Comp. Total | 58730 | 60629 | 61369 | 59538 | 59538 | 58486 | 58524 | 58463 |
| Perda Total | 62 | 19 | 3 | 2 | 2 | 1 | 0 | 1 |
| Ret. Total | 2408 | 4350 | 5106 | 3276 | 3276 | 2225 | 2264 | 2202 |
| PPequena | 5 | 9 | 1 | 2 | 2 | 1 | 0 | 1 |
| PNPequena | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Retalho | 1 | 5 | 9 | 2 | 2 | 2 | 2 | 4 |
| Solução | AC | IND | IND | AC | AC | AC | AC | IND |

Pela Tabela 6.2, observamos que as heurísticas RAG_A utilizam o menor comprimento dos objetos em estoque para atender a demanda e, dentre estas heurísticas, a versão 2 não gerou perdas no plano de corte, porém, o retalho ficou concentrado em dois objetos. A heurística RAG_A versão 3, mesmo com uma perda muito pequena teve a solução classificada como *indesejável* devido a quantidade de retalhos que deve retornar ao estoque. Esta mesma observação é válida para as heurísticas construtivas FFD_A e $Gulosa_A$. Com relação ao algoritmo CUT , observamos que o retalho ficou concentrado em um único objeto, porém, a perda gerada foi superior as heurísticas. De acordo com a Definição 3.1 nenhuma solução foi classificada como *ideal*, entretanto, as soluções das heurísticas construtivas e das residuais FFD_A , $Gulosa_A$ e RAG_A versão 3 são dominadas pelas heurísticas RAG_A versões 1 e 2.

Exemplo 2: O estoque é composto por $K = 20$ tipos de objetos com comprimentos entre 2200 cm e 6000 cm e disponibilidade de uma unidade para cada tipo. A demanda possui 5 tipos de itens ($m = 5$), cujos dados estão na Tabela 6.3.

Tabela 6.3: Dados do exemplo 2: Itens

| Item | Comprimento (cm) | Demanda |
|------|------------------|---------|
| 1 | 235 | 4 |
| 2 | 200 | 51 |
| 3 | 347 | 42 |
| 4 | 471 | 16 |
| 5 | 274 | 37 |

Para este exemplo, toda sobra superior a 200 cm é considerada retalho.

Tabela 6.4: Solução do Exemplo 2

| | <i>CUT</i> | Construtiva | | Residual | | | | |
|-------------|------------|------------------------|---------------------------|------------------------|---------------------------|--------------------------|--------------------------|--------------------------|
| | | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>RAG_A 1</i> | <i>RAG_A 2</i> | <i>RAG_A 3</i> |
| Obj. Cort. | 11 | 13 | 12 | 10 | 11 | 10 | 10 | 10 |
| Comp. Total | 44136 | 48506 | 46104 | 43803 | 46234 | 45245 | 45245 | 46507 |
| Perda Total | 5 | 8 | 1 | 3 | 3 | 0 | 0 | 0 |
| Ret. Total | 743 | 5110 | 2715 | 412 | 2852 | 1857 | 1857 | 3119 |
| PPequena | 3 | 3 | 1 | 2 | 1 | 0 | 0 | 0 |
| PNPequena | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Retalho | 1 | 7 | 4 | 1 | 2 | 1 | 1 | 2 |
| Solução | AC | IND | IND | AC | AC | ID | ID | AC |

Pela Tabela 6.4, observamos que as heurísticas RAG_A - versões 1 e 2 utilizam menos objetos que o algoritmo CUT para atender a demanda, porém maior comprimento, ou seja, as heurísticas utilizam os objetos de comprimentos maiores comparadas ao algoritmo CUT . Similar a este algoritmo, as heurísticas RAG_A - versões 1 e 2 concentram a sobra em um único objeto. Com relação às perdas, temos que estas duas heurísticas não geram perdas e são superiores ao algoritmo CUT com relação a este critério. Para este exemplo, utilizando os parâmetros definido a priori e a Definição 3.1 temos que as heurísticas RAG_A versões 1 e 2 são classificadas como *ideais* pois satisfazem os critérios estabelecidos.

Exemplo 3: O estoque possui $K = 20$ tipos de objetos com comprimentos entre 2100 cm e 5000 cm e disponibilidade de uma unidade para cada tipo. A demanda é composta por 5 tipos de itens cujos dados estão na Tabela 6.5.

Tabela 6.5: Dados do exemplo 3: Itens

| Item | Comprimento (cm) | Demanda |
|-------------|-------------------------|----------------|
| 1 | 549 | 39 |
| 2 | 433 | 27 |
| 3 | 207 | 43 |
| 4 | 308 | 39 |
| 5 | 583 | 2 |

Para este exemplo, temos que toda sobra superior a 207 cm é considerada retalho.

Tabela 6.6: Solução do Exemplo 3

| | <i>CUT</i> | Construtiva | | Residual | | | | |
|-------------|------------|------------------------|---------------------------|------------------------|---------------------------|--------------------------|--------------------------|--------------------------|
| | | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>RAG_A 1</i> | <i>RAG_A 2</i> | <i>RAG_A 3</i> |
| Obj. Cort. | 15 | 16 | 16 | 15 | 15 | 15 | 16 | 14 |
| Comp. Total | 56302 | 58356 | 57100 | 56256 | 56256 | 57554 | 58159 | 56395 |
| Perda Total | 104 | 17 | 17 | 9 | 9 | 6 | 3 | 7 |
| Ret. Total | 1017 | 3158 | 1902 | 1066 | 1066 | 2367 | 2972 | 1207 |
| PPequena | 7 | 8 | 8 | 5 | 5 | 4 | 4 | 5 |
| PNPequena | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Retalho | 7 | 3 | 2 | 1 | 1 | 2 | 3 | 2 |
| Solução | IND | IND | AC | AC | AC | AC | AC | AC |

Neste exemplo, embora as heurísticas propostas para resolver o aproveitamento de sobras sejam classificadas como aceitáveis (com exceção da heurística construtiva FFD_A), as heurísticas Residuais $Gulosa_A$ e FFD_A apresentam uma solução superior as demais heurísticas e ao algoritmo CUT , considerando os critérios: número de objetos cortados com perda não tão pequena e retalhos.

Nos próximos exemplos, temos em estoque $K = 90$ tipos de objetos com comprimentos entre 3000 cm e 9000 cm. A disponibilidade é de apenas uma unidade para cada tipo de objeto.

Exemplo 4: A demanda possui 15 tipos de itens, cujos dados estão na Tabela 6.7.

Tabela 6.7: Dados do exemplo 4: Itens

| Item | Comprimento (cm) | Demanda |
|------|------------------|---------|
| 1 | 569 | 34 |
| 2 | 718 | 26 |
| 3 | 520 | 25 |
| 4 | 540 | 12 |
| 5 | 492 | 30 |
| 6 | 547 | 2 |
| 7 | 632 | 6 |
| 8 | 430 | 36 |
| 9 | 750 | 7 |
| 10 | 387 | 20 |
| 11 | 804 | 3 |
| 12 | 389 | 32 |
| 13 | 835 | 18 |
| 14 | 684 | 39 |
| 15 | 687 | 10 |

Para este exemplo, $\delta = 387$ cm.

Tabela 6.8: Solução do Exemplo 4

| | <i>CUT</i> | Construtiva | | Residual | | | | |
|-------------|------------|------------------------|---------------------------|------------------------|---------------------------|--------------------------|--------------------------|--------------------------|
| | | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>RAG_A 1</i> | <i>RAG_A 2</i> | <i>RAG_A 3</i> |
| Obj. Cort. | 27 | 29 | 30 | 26 | 25 | 22 | 22 | 22 |
| Comp. Total | 170504 | 170343 | 173814 | 174176 | 173896 | 172114 | 172114 | 170989 |
| Perda Total | 2 | 14 | 0 | 5 | 0 | 0 | 0 | 0 |
| Ret. Total | 1456 | 1283 | 4768 | 5125 | 4850 | 3098 | 3098 | 1943 |
| PPequena | 2 | 9 | 0 | 4 | 0 | 0 | 0 | 0 |
| PNPequena | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Retalho | 1 | 1 | 6 | 2 | 3 | 1 | 1 | 1 |
| Solução | ID | AC | IND | AC | AC | ID | ID | ID |

Com 90 tipos de objetos em estoque, observamos pela Tabela 6.8 que as soluções apresentadas pelas heurísticas Residuais RAG_A - versões 1, 2 e 3 e pelo algoritmo *CUT* foram classificadas como *ideais* de acordo com a definição 3.1. As demais heurísticas, embora apresentem perdas baixas, foram classificadas como *aceitáveis* devido a quantidade de retalhos gerados.

Exemplo 5: A demanda para este exemplo é de 15 tipos de itens, cujos dados estão apresentados na Tabela 6.9.

Tabela 6.9: Dados do exemplo 5: Itens

| Item | Comprimento (cm) | Demanda |
|-------------|-------------------------|----------------|
| 1 | 548 | 36 |
| 2 | 492 | 45 |
| 3 | 641 | 33 |
| 4 | 414 | 24 |
| 5 | 303 | 46 |
| 6 | 861 | 16 |
| 7 | 498 | 23 |
| 8 | 748 | 23 |
| 9 | 733 | 39 |
| 10 | 658 | 16 |
| 11 | 570 | 39 |
| 12 | 898 | 21 |
| 13 | 327 | 47 |
| 14 | 801 | 40 |
| 15 | 471 | 32 |

Para este exemplo, o comprimento aceitável para o retalho deve ser superior a $\delta = 303$ cm.

Tabela 6.10: Solução do Exemplo 5

| | Construtiva | | | Residual | | | | |
|-------------|-------------|------------------------|---------------------------|------------------------|---------------------------|--------------------------|--------------------------|--------------------------|
| | <i>CUT</i> | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>RAG_A 1</i> | <i>RAG_A 2</i> | <i>RAG_A 3</i> |
| Obj. Cort. | 37 | 45 | 45 | 37 | 38 | 34 | 34 | 34 |
| Comp. Total | 255152 | 275154 | 273581 | 274071 | 278766 | 272648 | 272648 | 273267 |
| Perda Total | 3 | 5 | 0 | 1 | 0 | 0 | 0 | 0 |
| Ret. Total | 2366 | 3136 | 1568 | 2057 | 6753 | 635 | 635 | 1254 |
| PPequena | 2 | 5 | 0 | 1 | 0 | 0 | 0 | 0 |
| PNPequena | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Retalho | 1 | 2 | 3 | 1 | 2 | 1 | 1 | 2 |
| Solução | ID | ID | AC | ID | ID | ID | ID | ID |

Pela Tabela 6.10, podemos observar que as soluções apresentadas pelas heurísticas RAG_A versões 1 e 2 dominam as soluções das demais heurísticas e do algoritmo CUT , com relação a perda gerada e a quantidade de retalhos. Entretanto uma outra classificação poderia ser feita se o comprimento total de objetos utilizados para atender a demanda fosse analisado.

Exemplo 6: Neste exemplo, a demanda possui 15 tipos de itens, cujos dados estão apresentados na Tabela 6.11.

Tabela 6.11: Dados do exemplo 6: Itens

| Item | Comprimento (cm) | Demanda |
|------|------------------|---------|
| 1 | 861 | 3 |
| 2 | 359 | 37 |
| 3 | 312 | 52 |
| 4 | 826 | 31 |
| 5 | 598 | 35 |
| 6 | 738 | 42 |
| 7 | 538 | 29 |
| 8 | 817 | 15 |
| 9 | 661 | 33 |
| 10 | 678 | 14 |
| 11 | 642 | 35 |
| 12 | 670 | 44 |
| 13 | 899 | 12 |
| 14 | 525 | 34 |
| 15 | 510 | 34 |

Para este exemplo, o limite inferior para o retalho é 312 cm, ou seja, toda sobra superior a 312 cm é considerada retalho.

Tabela 6.12: Solução do Exemplo 6

| | <i>CUT</i> | Construtiva | | Residual | | | | |
|-------------|------------|------------------------|---------------------------|------------------------|---------------------------|--------------------------|--------------------------|--------------------------|
| | | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>RAG_A 1</i> | <i>RAG_A 2</i> | <i>RAG_A 3</i> |
| Obj. Cort. | 43 | 46 | 46 | 39 | 40 | 35 | 35 | 34 |
| Comp. Total | 268793 | 271567 | 271806 | 267050 | 271498 | 269699 | 269699 | 268702 |
| Perda Total | 1 | 16 | 0 | 3 | 0 | 0 | 0 | 0 |
| Ret. Total | 2080 | 4839 | 5094 | 335 | 4786 | 2987 | 2987 | 1990 |
| PPequena | 1 | 13 | 0 | 3 | 0 | 0 | 0 | 0 |
| PNPequena | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Retalho | 1 | 2 | 4 | 1 | 2 | 1 | 1 | 1 |
| Solução | ID | AC | AC | ID | ID | ID | ID | ID |

Pela Tabela 6.12, observamos que, apesar das heurísticas residuais e do algoritmo *CUT* serem classificados como *ideais*, as heurísticas *RAG_A* versões 1, 2 e 3 não geram perdas e, dentre elas, a heurística *RAG_A* versão 3 além de ser classificada como *ideal* utiliza o menor comprimento de objetos para atender a demanda.

Exemplo 7: A demanda possui 15 tipos de itens. O comprimento de cada item $i, i = 1, \dots, 15$ e suas respectivas demandas estão apresentados na Tabela 6.13.

Tabela 6.13: Dados do exemplo 7: Itens

| Item | Comprimento (cm) | Demanda |
|-------------|-------------------------|----------------|
| 1 | 858 | 33 |
| 2 | 521 | 22 |
| 3 | 885 | 28 |
| 4 | 650 | 43 |
| 5 | 741 | 45 |
| 6 | 578 | 10 |
| 7 | 476 | 21 |
| 8 | 467 | 17 |
| 9 | 427 | 47 |
| 10 | 876 | 36 |
| 11 | 672 | 16 |
| 12 | 467 | 27 |
| 13 | 655 | 48 |
| 14 | 887 | 47 |
| 15 | 716 | 10 |

Com esta demanda, o comprimento aceitável para o retalho deve ser superior a 427 cm.

Tabela 6.14: Solução do Exemplo 7

| | <i>CUT</i> | Construtiva | | Residual | | | | |
|-------------|------------|------------------------|---------------------------|------------------------|---------------------------|--------------------------|--------------------------|--------------------------|
| | | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>RAG_A 1</i> | <i>RAG_A 2</i> | <i>RAG_A 3</i> |
| Obj. Cort. | 50 | 55 | 57 | 46 | 46 | 43 | 43 | 44 |
| Comp. Total | 305752 | 305937 | 320098 | 306089 | 306089 | 305428 | 305428 | 309341 |
| Perda Total | 8 | 15 | 4 | 6 | 1 | 0 | 0 | 1 |
| Ret. Total | 923 | 1101 | 15273 | 1262 | 1267 | 607 | 607 | 4519 |
| PPequena | 8 | 12 | 3 | 5 | 1 | 0 | 0 | 1 |
| PNPequena | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Retalho | 1 | 2 | 14 | 1 | 1 | 1 | 1 | 2 |
| Solução | AC | AC | IND | ID | ID | ID | ID | ID |

Pela Tabela 6.14, observamos que a solução da heurística construtiva $Gulosa_A$ apresenta uma perda muito pequena, porém é classificada como *indesejável* por gerar uma quantidade muito elevada de retalhos. As heurísticas residuais foram classificadas como *ideais*, porém, dentre elas, as heurísticas RAG_A - versões 1 e 2 apresentaram soluções superiores por não gerarem perdas.

Exemplo 8: A demanda é composta por 15 tipos de itens, cujos dados referentes a comprimento e demanda estão apresentados na Tabela 6.15.

Tabela 6.15: Dados do exemplo 8: Itens

| Item | Comprimento (cm) | Demanda |
|-------------|-------------------------|----------------|
| 1 | 757 | 44 |
| 2 | 307 | 29 |
| 3 | 665 | 53 |
| 4 | 805 | 23 |
| 5 | 433 | 11 |
| 6 | 457 | 62 |
| 7 | 387 | 42 |
| 8 | 541 | 14 |
| 9 | 308 | 2 |
| 10 | 506 | 47 |
| 11 | 828 | 32 |
| 12 | 822 | 23 |
| 13 | 442 | 35 |
| 14 | 779 | 15 |
| 15 | 719 | 18 |

O comprimento do retalho deve ser superior a 307 cm.

Tabela 6.16: Solução do Exemplo 8

| | <i>CUT</i> | Construtiva | | Residual | | | | |
|-------------|------------|------------------------|---------------------------|------------------------|---------------------------|--------------------------|--------------------------|--------------------------|
| | | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>FFD_A</i> | <i>Gulosa_A</i> | <i>RAG_A 1</i> | <i>RAG_A 2</i> | <i>RAG_A 3</i> |
| Obj. Cort. | 43 | 45 | 45 | 40 | 40 | 36 | 36 | 36 |
| Comp. Total | 263565 | 265248 | 266399 | 267600 | 266988 | 265013 | 265013 | 264127 |
| Perda Total | 5 | 13 | 1 | 1 | 0 | 0 | 0 | 0 |
| Ret. Total | 767 | 2442 | 3605 | 4806 | 4195 | 2220 | 2220 | 1334 |
| PPequena | 2 | 11 | 1 | 1 | 0 | 0 | 0 | 0 |
| PNPequena | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Retalho | 1 | 3 | 6 | 3 | 2 | 1 | 1 | 1 |
| Solução | ID | AC | IND | AC | ID | ID | ID | ID |

Para este exemplo, observamos que as heurísticas RAG_A - versões 1, 2 e 3 utilizam maior comprimento de objetos em estoque para atender a demanda quando comparadas ao algoritmo CUT, porém suas soluções não apresentam perdas.

De modo geral, observamos que em todos os exemplos apresentados, o desempenho das heurísticas RAG_A foi melhor ou equivalente ao desempenho do algoritmo CUT de acordo com a Definição 3.1.

Os próximos exemplos foram retirados de Abuabara e Morabito (2008), cujo trabalho é baseado nos modelos propostos por Gradisar *et al.* (1997). Os exemplos foram retirados de carteiras de pedidos de uma pequena indústria brasileira de aeronaves agrícolas que realiza o corte de tubos metálicos na fabricação de suas aeronaves, cujas estruturas são formadas por pórticos treliçados.

Os exemplos apresentados, característicos de empresas pequenas, não são classificados conforme a Definição 3.1, pois os valores $\varepsilon_1 = 0,03$ e $\varepsilon_2 = 0,1$ estabelecidos anteriormente não são apropriados devido a quantidade de objetos utilizados para atender a demanda ser pequena (baixa demanda) e desta forma, o limitante obtido para quantidade de objetos cortados com perda pequena, não tão pequena e retalhos é muito pequeno. Para estes exemplos, a escolha da melhor solução de acordo com a Definição 3.1 pode ser feita depois que as quantidades *poucos*, *muito poucos* e *vários* forem definidas pelo decisor.

Nos exemplos a seguir, temos um único tipo de objeto em estoque, ou seja, $e_k =$

$1, k = 1$. Como o estoque de objetos é composto apenas por objetos padronizados, definimos somente o parâmetro θ , sendo que $\theta = 0,005$. O tamanho mínimo para os retalhos é $\delta = \min\{\ell_i, i = 1, \dots, m\}$.

Para estes exemplos, além das soluções das heurísticas de aproveitamento também apresentamos as soluções das respectivas versões clássicas. As soluções das heurísticas RAG e RAG_A - versões 2 e 3 não são apresentadas pelo fato destas heurísticas gerarem soluções similares as soluções das heurísticas RAG e RAG_A - versão 1.

Exemplo 9: O estoque é composto de 10 objetos com 3000 cm. A demanda possui 5 tipos de itens, cujos dados estão na Tabela 6.17.

Tabela 6.17: Dados do exemplo 9: Itens

| Item | Comprimento (cm) | Demanda |
|------|------------------|---------|
| 1 | 250 | 2 |
| 2 | 273 | 2 |
| 3 | 285 | 4 |
| 4 | 525 | 4 |
| 5 | 1380 | 4 |

Qualquer sobras de material em um padrão de corte superior a 250 cm é considerada retalho.

Tabela 6.18: Solução do Exemplo 9

| | Construtiva | | | | Residual | | | | | |
|-------------|-------------|------------------|--------|---------------------|----------|------------------|--------|---------------------|-------|--------------------|
| | FFD | FFD _A | Gulosa | Gulosa _A | FFD | FFD _A | Gulosa | Gulosa _A | RAG 1 | RAG _A 1 |
| Obj. Cort. | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 |
| Comp. Total | 12000 | 15000 | 12000 | 15000 | 12000 | 15000 | 12000 | 15000 | 12000 | 15000 |
| Perda Total | 525 | 0 | 240 | 0 | 240 | 0 | 240 | 0 | 240 | 0 |
| Ret. Total | 1669 | 5194 | 1954 | 5194 | 1954 | 5194 | 1954 | 5194 | 1954 | 5194 |
| PPequena | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PNPequena | 3 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| Retalho | 1 | 3 | 1 | 3 | 1 | 3 | 1 | 3 | 1 | 3 |

Pela Tabela 6.18 observamos que a heurística RAG_A 1 não gera perdas, porém, apresenta maior quantidade de retalhos quando comparada com sua versão original. Isto

ocorre pelo fato das heurísticas de aproveitamento eliminarem, na maioria dos problemas, perdas não tão pequenas durante o processo de corte.

Neste exemplo e nos seguintes, nos deparamos com soluções em que temos perdas de material e um único retalho para estoque, ou soluções sem perdas mas com um grande número de retalhos. Esta é uma situação típica que ocorre quando resolvemos estes problemas, pois raramente obtemos uma solução que é boa considerando todos os critérios. Desta forma, a escolha da melhor solução deve ser feita pelo decisor, que está integrado na realidade da empresa.

Exemplo 10: O estoque é composto por 10 objetos de 6000 cm e a demanda de 4 tipos de itens, ou seja, $m = 4$. O comprimento de cada item $i, i = 1, \dots, 4$ e suas respectivas demandas estão na Tabela 6.19.

Tabela 6.19: Dados do exemplo 10: Itens

| Item | Comprimento (cm) | Demanda |
|------|------------------|---------|
| 1 | 370 | 5 |
| 2 | 905 | 5 |
| 3 | 910 | 5 |
| 4 | 930 | 5 |

Com esta demanda, o comprimento aceitável para o retalho deve ser superior a 370 cm.

Tabela 6.20: Solução do Exemplo 10

| | Construtiva | | | | Residual | | | | | |
|-------------|-------------|------------------------|---------------|---------------------------|------------|------------------------|---------------|---------------------------|--------------|--------------------------|
| | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> |
| Obj. Cort. | 3 | 4 | 3 | 3 | 3 | 4 | 3 | 3 | 3 | 3 |
| Comp. Total | 18000 | 24000 | 18000 | 18000 | 18000 | 24000 | 18000 | 18000 | 18000 | 18000 |
| Perda Total | 250 | 0 | 250 | 0 | 250 | 0 | 250 | 0 | 515 | 150 |
| Ret. Total | 2175 | 8425 | 2175 | 2425 | 2175 | 8425 | 2175 | 2425 | 1910 | 2275 |
| PPequena | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PNPequena | 2 | 0 | 2 | 0 | 2 | 0 | 2 | 0 | 2 | 1 |
| Retalho | 1 | 4 | 1 | 3 | 1 | 4 | 1 | 3 | 1 | 2 |

Como podemos observar, as heurísticas clássicas geram apenas um retalho para o estoque, porém todas apresentam perdas maiores quando comparadas com as heurísticas

de aproveitamento. Na heurística $RAG_A 1$ não foi possível eliminar totalmente as perdas não tão pequenas, porém a quantidade de retalhos que retornam ao estoque é menor que nas demais heurísticas de aproveitamento. Como observado anteriormente, a escolha da melhor solução não é trivial, pois envolve a análise simultânea de várias características. A opção de gerar apenas um retalho leva a perdas significativamente maiores, enquanto que soluções com perdas nulas têm um número significativamente maior de retalhos. Novamente, o exemplo evidencia o conflito entre objetivos e assim, o decisor deve fazer sua escolha.

Exemplo 11: O estoque é composto de 15 objetos com 6000 cm. A demanda é composta por 7 tipos de itens, cujos dados estão na Tabela 6.21.

Tabela 6.21: Dados do exemplo 11: Itens

| Item | Comprimento (cm) | Demanda |
|------|------------------|---------|
| 1 | 350 | 12 |
| 2 | 540 | 3 |
| 3 | 705 | 3 |
| 4 | 735 | 6 |
| 5 | 760 | 6 |
| 6 | 890 | 6 |
| 7 | 900 | 3 |

Qualquer sobra com comprimento superior a 350 cm é considerada retalho.

Tabela 6.22: Solução do Exemplo 11

| | Construtiva | | | | Residual | | | | | |
|-------------|-------------|---------|----------|------------|----------|---------|----------|------------|---------|-----------|
| | FFD | FFD_A | $Gulosa$ | $Gulosa_A$ | FFD | FFD_A | $Gulosa$ | $Gulosa_A$ | $RAG 1$ | $RAG_A 1$ |
| Obj. Cort. | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| Comp. Total | 30000 | 30000 | 30000 | 30000 | 30000 | 30000 | 30000 | 30000 | 30000 | 30000 |
| Perda Total | 455 | 30 | 0 | 0 | 455 | 0 | 140 | 0 | 305 | 0 |
| Ret. Total | 4600 | 5025 | 5055 | 5055 | 4600 | 5055 | 4915 | 5055 | 4750 | 5055 |
| PPequena | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PNPequena | 3 | 0 | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 0 |
| Retalho | 1 | 2 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |

Pela Tabela 6.22, observamos que para satisfazer toda a demanda apenas um pequeno número de objetos foi utilizado. Como nos exemplos anteriores, as heurísticas de

aproveitamento geram soluções com perdas reduzidas e com maior quantidade de retalhos para o estoque, enquanto que as heurísticas clássicas geram maior perda e menor quantidade de retalhos.

Para estes exemplos, soluções baseadas nos modelos matemáticos de Abuabara e Morabito (2008) podem ser obtidas, sendo que nesses modelos, restrições no número de retalhos que podem ser gerados são impostas. As soluções obtidas por esses modelos são, em geral, similares às soluções obtidas utilizando as heurísticas modificadas para resolver o aproveitamento de sobras (em 9 exemplares, dos 13 apresentados em Abuabara e Morabito (2008), as soluções obtidas pela heurística RAG_A 2 foram equivalentes às soluções do modelo matemático). Entretanto, quando consideramos problemas com grande variedade de tipos de objetos em estoque e alta disponibilidade, bem como grande variedade de tipos de itens demandados em grandes quantidades, os modelos matemáticos apresentam um grande número de variáveis e restrições e softwares avançados como CPLEX, freqüentemente não são capazes de produzir boas soluções.

6.1.2 Resultados de exemplares gerados aleatoriamente

Nesta seção apresentamos os resultados computacionais de exemplares gerados aleatoriamente conforme o gerador de problemas descrito em Cherri *et al.* (2009). Os dados para os exemplares e para o gerador são:

- *Número de tipos de objetos padronizados:* $\bar{k} = 2$;
- *Número de tipos de objetos não padronizados:* $k = 3, 5$ e 7 ;
- *Número total de tipos de objetos em estoque:* $K = 5, 7$ e 9 ;
- *Número de tipos de itens:* $m = 10, 20$ e 40 ;
- *Disponibilidades dos tipos de objetos padronizados:* $e_1 = e_2 = 100$ objetos;
- *Disponibilidades dos tipos de objetos não padronizados:* $e_k, k = 3, \dots, K$, são gerados aleatoriamente no intervalo $[1, 10]$;
- *Comprimentos dos objetos padronizados:* $L_1 = 1000$ e $L_2 = 1100$;

- *Comprimentos dos itens:* Os itens de comprimento ℓ_i são gerados aleatoriamente no intervalo $[v_1L, v_2L]$, em que L é o valor médio de $L_k, k = 1, 2, v_1 = 0,01, v_2 = 0,2$ e $0,8$. Combinando estes valores, classes de exemplos são geradas. Para itens pequenos consideramos $v_2 = 0,2$ e para itens médios $v_2 = 0,8$. Obviamente, itens pequenos podem ser definidos utilizando outros parâmetros;
- *Tamanho aceitável para um retalho:* $\left[\delta = \frac{1}{m} \sum_{i=1}^m \ell_i \right]$;
- *Comprimentos dos objetos não padronizados:* $L_k, k = 3, \dots, K$ são gerados aleatoriamente no intervalo de $[\delta, \frac{L_1}{2}]$;
- *Demanda:* d_i é gerada aleatoriamente no intervalo $\left[0,02 \frac{\sum_{k=1}^K e_k L_k}{\sum_{i=1}^m \ell_i}, \frac{\sum_{k=1}^K e_k L_k}{\sum_{i=1}^m \ell_i} \right], i = 1, \dots, m$. Combinado estes valores, garantimos que o comprimento total dos itens demandados não exceda o comprimento total dos objetos em estoque;
- *Tamanho máximo da perda pequena nos objetos padronizados:* $0,005L_k, k = 1, 2$, ou seja, $\theta = 0,005$;
- *Tamanho máximo da perda pequena nos objetos não padronizados:* $0,05L_k, k = 3, \dots, K$, ou seja, $\beta = 0,05$;

Nos exemplares gerados, como não priorizamos o corte dos objetos não padronizados, definimos uma porcentagem de perda aceitável maior para estes objetos, aumentando a possibilidade de padrões de corte gerados por objetos não padronizados serem aceitos.

Para os testes computacionais, 16 classes de exemplos foram geradas combinando os parâmetros $K = (5, 7 \text{ ou } 9)$, $m = (10 \text{ e } 20 \text{ ou } 40)$, $v_2 = 0,2$ (itens pequenos: P) e $v_2 = 0,8$ (itens médios: M). Na tabela 6.23 estão descritas as 16 classes de exemplos gerados. Por exemplo, a classe 1 é formada por exemplares com $K = 5$ tipos de objetos, que devem ser cortados para a produção de $m = 10$ tipos de itens, cujos comprimentos são gerados aleatoriamente no intervalo $[0,01L, 0,2L]$, com $L = (L_1 + L_2)/2 = 1050$. A

classe 2 tem o mesmo tamanho da classe 1 em termos de quantidade de tipos de objetos e de itens, porém os itens são gerados aleatoriamente no intervalo $[0, 01L \ 0, 8L]$.

Tabela 6.23: Descrição das classes

| Classe | Parâmetros | | |
|--------|------------|-----|-------|
| | K | m | Itens |
| 1 | 5 | 10 | P |
| 2 | 5 | 10 | M |
| 3 | 5 | 20 | P |
| 4 | 5 | 20 | M |
| 5 | 5 | 40 | P |
| 6 | 5 | 40 | M |
| 7 | 7 | 10 | P |
| 8 | 7 | 10 | M |
| 9 | 7 | 20 | P |
| 10 | 7 | 20 | M |
| 11 | 7 | 40 | P |
| 12 | 7 | 40 | M |
| 13 | 9 | 10 | P |
| 14 | 9 | 10 | M |
| 15 | 9 | 20 | P |
| 16 | 9 | 20 | M |

Para cada classe descrita na Tabela 6.23, 20 exemplares foram gerados. A média dos resultados obtidos com testes computacionais realizados para os 320 exemplares (16 classes \times 20 exemplares) são apresentados nas tabelas a seguir. Nestas tabelas, temos:

- i. Média do número total (Número) e comprimento total (Comp.) de objetos não padronizados e padronizados utilizados (Tabelas 6.24 e 6.25);
- ii. Média da perda total (Perda), média do comprimento total da perda nos objetos padronizados (Padroniz.) e não padronizados (Não Padroniz.) (Tabela 6.26);
- iii. Média do comprimento total dos retalhos (Retalho), média do comprimento total dos retalhos nos objetos padronizados (Padroniz.) e não padronizados (Não Padroniz.) (Tabela 6.27);
- iv. Número médio de objetos cortados com: retalho, perda pequena e perda não tão pequena (Tabela 6.28);

- v. Porcentagem média de objetos cortados com: retalho, perda pequena e perda não tão pequena (Tabela 6.29);
- vi. Tempo médio de execução (Tabela 6.30);
- vii. Classificação geral das heurísticas (Tabela 6.31).

Em cada linha das tabelas a seguir, os maiores e menores valores estão em **negrito** e *itálico*, respectivamente. Os detalhes dos resultados de cada tabela estão no Apêndice A.

Tabela 6.24: Número médio e comprimento médio total de objetos não padronizados usados

| | | Número de Padrões de Corte com Objetos Retalhos | | | | | | | | | | | | | | |
|--------|---------------|---|------------|------------------------|---------------|---------------------------|------------|------------------------|---------------|---------------------------|--------------|--------------------------|--------------|--------------------------|--------------|--------------------------|
| | | Construtiva | | | | Residual | | | | | | | | | | |
| | | <i>COLA</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> | <i>RAG 2</i> | <i>RAG_A 2</i> | <i>RAG 3</i> | <i>RAG_A 3</i> |
| Número | 28,1 | 11,0 | 8,6 | 8,4 | 9,6 | 9,1 | 5,8 | 5,0 | 4,8 | 3,7 | 4,3 | 4,0 | 4,4 | 3,5 | 4,5 | |
| Comp. | 7065,3 | 2445,0 | 1820,4 | 1865,5 | 2510,3 | 1827,0 | 1058,1 | 911,9 | 1014,7 | 841,3 | 1025,4 | 884,8 | 1041,6 | 800,9 | 1052,5 | |

127

Tabela 6.25: Número médio e comprimento médio total de objetos padronizados usados

| | | Número de Padrões de Corte com Objetos Retalhos | | | | | | | | | | | | | | |
|--------|----------|---|------------|------------------------|-----------------|---------------------------|------------|------------------------|---------------|---------------------------|--------------|--------------------------|--------------|--------------------------|--------------|--------------------------|
| | | Construtiva | | | | Residual | | | | | | | | | | |
| | | <i>COLA</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> | <i>RAG 2</i> | <i>RAG_A 2</i> | <i>RAG 3</i> | <i>RAG_A 3</i> |
| Número | 104,0 | 105,8 | 108,5 | 109,0 | 110,9 | 101,8 | 102,7 | 101,6 | 102,9 | 102,3 | 102,6 | 102,2 | 102,5 | 102,3 | 102,6 | |
| Comp. | 105120,9 | 109978,1 | 111161,9 | 110649,7 | 113147,2 | 109886,3 | 110724,1 | 110720,9 | 110816,9 | 110612,5 | 110932,2 | 110589,7 | 110881,9 | 110655,0 | 110926,9 | |

Diferentemente dos procedimentos heurísticos que consideram o aproveitamento de sobras, com o algoritmo COLA os objetos são ordenados de acordo com seu comprimento de modo não decrescente e, para cada objeto nesta seqüência, um padrão de corte é construído. Assim, o algoritmo COLA prioriza o uso de objetos não padronizados e utiliza o maior número e comprimento destes objetos na solução (Tabela 6.24). Entre as heurísticas originais e as modificadas, as heurísticas construtivas FFD e Gulosa_A são as que mais utilizam objetos não padronizados, enquanto que as heurísticas RAG - versões 1, 2 e 3 são as que menos utilizam esses objetos.

Devemos lembrar que o uso de objetos não padronizados não foi prioridade nestas heurísticas desenvolvidas (prioridades no corte de objetos não padronizados estão na Seção 6.2). O uso destes objetos foi estimulado com uma tolerância maior para o conceito de perda pequena. Caso seja relevante a redução do estoque de objetos não padronizados, as heurísticas podem ser revisadas, naturalmente em detrimento de outras características desejáveis.

Pela Tabela 6.25 observamos que a heurística construtiva Gulosa_A utiliza a maior quantidade e comprimento total de objetos padronizados. Isto ocorre devido a facilidade que a heurística apresenta em alocar (combinar) itens em objetos com comprimentos maiores. O algoritmo COLA, embora não utilize a menor quantidade, utiliza o menor comprimento de objetos padronizados, pois pela ordenação estabelecida, utiliza inicialmente os objetos com comprimentos menores.

Tabela 6.26: Perda total média, comprimento médio da perda nos objetos padronizados e não padronizados

| | | Número de Padrões de Corte com Objetos Retalhos | | | | | | | | | | | | | | |
|---------------|-------|---|------------|------------------------|---------------|---------------------------|------------|------------------------|---------------|---------------------------|--------------|--------------------------|--------------|--------------------------|--------------|--------------------------|
| | | Construtiva | | | | Residual | | | | | | | | | | |
| | | <i>COLA</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> | <i>RAG 2</i> | <i>RAG_A 2</i> | <i>RAG 3</i> | <i>RAG_A 3</i> |
| Perda | 126,3 | 577,2 | 37,1 | 575,5 | 29,2 | 61,3 | 15,6 | 73,9 | 16,3 | 60,1 | 11,5 | 65,3 | 11,8 | 66,1 | 12,5 | |
| Padroniz. | 114,9 | 552,1 | 32,2 | 456,3 | 14,7 | 13,6 | 11,8 | 14,7 | 11,4 | 16,2 | 6,2 | 15,7 | 6,6 | 21,5 | 7,0 | |
| Não Padroniz. | 11,4 | 23,5 | 4,9 | 119,3 | 14,4 | 37,1 | 3,8 | 59,3 | 4,9 | 43,4 | 5,3 | 49,6 | 5,3 | 46,6 | 5,1 | |

129

Tabela 6.27: Comprimento médio total dos retalhos, comprimento médio dos retalhos nos objetos padronizados e não padronizados

| | | Número de Padrões de Corte com Objetos Retalhos | | | | | | | | | | | | | | |
|---------------|-------|---|------------|------------------------|---------------|---------------------------|------------|------------------------|---------------|---------------------------|--------------|--------------------------|--------------|--------------------------|--------------|--------------------------|
| | | Construtiva | | | | Residual | | | | | | | | | | |
| | | <i>COLA</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> | <i>RAG 2</i> | <i>RAG_A 2</i> | <i>RAG 3</i> | <i>RAG_A 3</i> |
| Retalho | 685,8 | 489,6 | 1592,6 | 652,5 | 4269,6 | 293,1 | 401,7 | 204,0 | 455,2 | 34,9 | 587,4 | 39,0 | 552,8 | 49,6 | 601,9 | |
| Padroniz. | 663,3 | 488,5 | 1489,2 | 492,2 | 3688,0 | 267,9 | 222,7 | 108,6 | 207,5 | 4,4 | 383,5 | 2,5 | 342,0 | 8,0 | 383,8 | |
| Não Padroniz. | 0,0 | 1,1 | 103,5 | 159,1 | 581,5 | 25,2 | 178,9 | 95,3 | 249,0 | 30,5 | 195,7 | 36,5 | 210,9 | 41,6 | 217,5 | |

Pela Tabela 6.26, observamos que as heurísticas residuais modificadas apresentam soluções melhores quando a perda é analisada. Um fato que devemos destacar, é que mesmo permitindo uma perda maior para os objetos não padronizados (uma maneira destes objetos terem preferência de corte em relação aos objetos padronizados) as soluções apresentadas pelas heurísticas de aproveitamento são, entre as heurísticas residuais, as que apresentam menores perdas. Considerando o algoritmo COLA, observamos que este algoritmo apresenta perdas maiores comparado com as heurísticas de aproveitamento de sobras. Naturalmente, a redução das perdas deve-se ao fato de que sobras grandes (i.e., retalhos) não são computadas como perdas, uma vez que tornam-se objetos não padronizados futuros. Entretanto, o aumento de objetos não padronizados futuros pode comprometer a qualidade da solução, conforme a Definição 3.1, tornando uma solução *indesejável*.

De acordo com a Definição 3.1 o comprimento total dos retalhos não qualifica uma solução como boa ou ruim, pois uma solução com retalho total grande pode ainda ser considerada *ideal* ou *aceitável*, desde que ele esteja concentrado em poucos objetos. As informações sobre o comprimento dos retalhos na Tabela 6.27 são interessantes para observar o efeito negativo (aumento de retalhos) decorrente da diminuição de perdas não tão pequenas. Por exemplo, a heurística construtiva FFD_A reduz a perda, em média, de 577,2 para 37,1 (Tabela 6.26), porém o comprimento total dos retalhos aumenta, em média, de 489,6 para 1592,6 (Tabela 6.27).

Pela Tabela 6.27 podemos observar que a heurística construtiva $Gulosa_A$ produz, em média, os maiores retalhos. Isto já era esperado, uma vez que esta heurística transforma todas as perdas indesejáveis em retalhos. O algoritmo COLA, devido a ordenação dos objetos, não gera retalhos nos objetos não padronizados, pois estes objetos são cortados no início do processo de corte, quando toda demanda ainda não foi atendida. Sendo assim, é provável que bons padrões sejam gerados para estes objetos.

Em termos da Definição 3.1, é mais importante a quantidade de objetos cortados com retalho do que o comprimento que eles apresentam. Note que a escolha do melhor procedimento heurístico para resolver o problema de corte com sobras aproveitáveis não é trivial uma vez que vários fatores devem ser analisados simultaneamente.

Tabela 6.28: Número médio de objetos cortados com: retalho, perda pequena e perda não tão pequena

| | Número de Padrões de Corte com Objetos Retalhos | | | | | | | | | | | | | | |
|----------------|---|-------------|------------------|--------|---------------------|----------|------------------|--------|---------------------|-------|--------------------|-------|--------------------|-------|--------------------|
| | COLA | Construtiva | | | | Residual | | | | | | | | | |
| | | FFD | FFD _A | Gulosa | Gulosa _A | FFD | FFD _A | Gulosa | Gulosa _A | RAG 1 | RAG _A 1 | RAG 2 | RAG _A 2 | RAG 3 | RAG _A 3 |
| Retalho | 3,4 | 1,0 | 5,6 | 3,9 | 17,6 | 0,7 | 1,8 | 0,9 | 2,5 | 0,3 | 2,6 | 0,3 | 2,6 | 0,5 | 2,8 |
| Perda Peq. | 9,0 | 45,7 | 16,6 | 8,5 | 8,9 | 6,7 | 3,5 | 3,3 | 3,4 | 3,1 | 3,2 | 3,1 | 3,2 | 3,3 | 3,6 |
| Perda Não Peq. | 4,3 | 22,5 | 0,0 | 11,4 | 0,0 | 1,9 | 0,6 | 2,2 | 0,6 | 1,8 | 0,1 | 2,0 | 0,1 | 2,0 | 0,1 |

Tabela 6.29: Porcentagem média de objetos cortados com: retalho, perda pequena e perda não tão pequena

| | Número de Padrões de Corte com Objetos Retalhos | | | | | | | | | | | | | | |
|----------------|---|-------------|------------------|--------|---------------------|----------|------------------|--------|---------------------|-------|--------------------|-------|--------------------|-------|--------------------|
| | COLA | Construtiva | | | | Residual | | | | | | | | | |
| | | FFD | FFD _A | Gulosa | Gulosa _A | FFD | FFD _A | Gulosa | Gulosa _A | RAG 1 | RAG _A 1 | RAG 2 | RAG _A 2 | RAG 3 | RAG _A 3 |
| Retalho | 2,6 | 0,9 | 4,8 | 3,3 | 14,6 | 0,6 | 1,7 | 0,9 | 2,3 | 0,3 | 2,4 | 0,3 | 2,4 | 0,5 | 2,6 |
| Perda Peq. | 6,8 | 39,2 | 14,2 | 7,2 | 7,4 | 6,0 | 3,2 | 3,1 | 3,2 | 2,9 | 3,0 | 2,9 | 3,0 | 3,1 | 3,4 |
| Perda Não Peq. | 3,3 | 19,3 | 0,0 | 9,7 | 0,0 | 1,7 | 0,6 | 2,0 | 0,6 | 1,7 | < 0,1 | 1,9 | < 0,1 | 1,9 | < 0,1 |

Pela Tabela 6.28 observamos que as heurísticas modificadas apresentam uma grande quantidade de objetos cortados com retalhos quando comparadas com suas versões originais. Em geral, isto ocorre porque os padrões de corte com perdas não tão pequenas são modificados por estas heurísticas. Veja, por exemplo, que a heurística RAG - versão 1 apresenta, em média, 0,3 objetos cortados com retalhos, enquanto que a sua versão para o aproveitamento de sobras, RAG_A - versão 1, apresenta 2,6 objetos cortados com retalhos. Observe que mesmo não eliminando todos os objetos não padronizados do estoque (Tabela 6.24) as heurísticas de aproveitamento geram, em média, uma quantidade menor de retalhos (Tabela 6.28) com relação ao que foi utilizado (com exceção da heurística construtiva $Gulosa_A$), ou seja, o estoque de objetos não padronizados (com exceção da heurística construtiva $Gulosa_A$) tende a diminuir.

Para estes exemplos gerados aleatoriamente, o algoritmo COLA apresentou, em média, uma grande quantidade de objetos cortados com retalhos, perdas pequenas (objetos com perdas nulas não são contabilizados) e perdas não tão pequenas quando comparado com os valores das heurísticas residuais.

Pela Tabela 6.29 observamos que as heurísticas originais apresentam menor porcentagem de objetos cortados com retalhos. As heurísticas modificadas para resolverem o problema de aproveitamento de sobras apresentam uma porcentagem maior de retalhos pelo fato de tentarem evitar perdas indesejáveis (perdas não tão pequenas). Este fato pode ser observado quando analisamos a porcentagem de objetos cortados com perda não tão pequena (linha Perda Não Peq. da Tabela 6.29), em que as heurísticas de aproveitamento apresentam solução superior às heurísticas originais (clássicas) e ao algoritmo COLA.

Na Tabela 6.30 apresentamos o tempo médio de execução para resolver os 320 exemplares gerados aleatoriamente. Observe que as heurísticas construtivas são muito mais rápidas que as heurísticas residuais, porém o tempo computacional é bastante razoável para o uso na prática de qualquer uma das heurísticas.

Para a classificação das heurísticas, conforme a Definição 3.1, relaxamos a posição da *solução ideal* de ‘nenhum’ objeto cortado com perda não tão pequena para no máximo 0,1%. Desta forma, observamos que as heurísticas RAG_A - versões 1, 2 e 3 foram classificadas como *ideais* enquanto que as demais heurísticas residuais e o algoritmo COLA foram classificados como *aceitáveis*. Como temos três heurísticas com *solução ideal*, podemos usar ainda outros critérios para desempate que podem ser definidos pelo decisor. Esse critério pode ser por exemplo, o comprimento da perda gerada, a quantidade de novos retalhos gerados, entre outros.

6.2 Problema de corte com sobras aproveitáveis e prioridade no uso de retalhos

Para avaliar o desempenho da heurística RAG_A^P realizamos uma simulação com vários períodos de tempo, em que a cada período as informações de estoque (objetos padronizados restantes e retalhos que foram gerados) para o problema no período seguinte são transferidas do período anterior. A cada período uma nova demanda é gerada aleatoriamente com itens regulares (itens que têm saída em todo período) e itens encomendados (não regulares). As informações referentes aos dados dos problemas são descritas a seguir.

Objetos

Consideramos dois tipos de objetos padronizados, $L_1 = 1000$ e $L_2 = 1100$, com disponibilidade suficiente para atender a demanda de todos os períodos.

Vetor de demanda geral

Para obtermos a demanda em cada período, geramos inicialmente um vetor (denominado de vetor $v^{inicial}$) com 50 tipos de itens com comprimento $\ell_i, i = 1, 2, \dots, 50$, os quais são gerados aleatoriamente no intervalo $[v_1L, v_2L]$, em que L é o valor médio do comprimento dos objetos padronizados. Os valores de v_1 e v_2 são definidos para 3 classes

de itens:

- *itens pequenos*: $v_1 = 0,01$ e $v_2 = 0,25$;
- *itens médios*: $v_1 = 0,01$ e $v_2 = 0,4$;
- *itens variados*: $v_1 = 0,01$, $v_2 = 0,25$ para $\ell_i, i = 1, \dots, 5$, $v_1 = 0,01$, $v_2 = 0,4$ para $\ell_i, i = 6, \dots, 10$ e $v_1 = 0,001$, $v_2 = 0,4$ para $\ell_i, i = 11, \dots, 50$;

Obviamente, classes de itens pequenos, médios e variados podem ser definidas por outros parâmetros. Para estas classes de itens definidas, os 10 primeiros itens são considerados itens regulares e são demandados em todos os períodos. As demandas destes itens são geradas aleatoriamente no intervalo $[200, 500]$. Para os itens não regulares, geramos aleatoriamente, em cada período, um valor inteiro m no intervalo $[10, 30]$ e selecionamos aleatoriamente m tipos de itens do vetor $v^{inicial}$ dentre os elementos 11 a 50. A demanda de cada um dos m itens selecionados é gerada aleatoriamente no intervalo $[1, 10]$. Assim, temos que o número de tipos de itens em cada período é dado por $(10 + m)$.

Limites para perda e retalho

- *Comprimento máximo da perda pequena para os objetos padronizados*: $\theta L_k, k = 1, 2$, em que $\theta = 0,005$ para itens pequenos e $\theta = 0,01$ para itens médios e variados;
- *Comprimento máximo da perda pequena para os objetos não padronizados (retalhos)*: $\beta = 5$, para itens pequenos e $\beta = 10$ para itens médios e variados;
- *Comprimento mínimo aceitável para o retalho*: $\delta =$ média do comprimento dos 10 primeiros itens demandados do vetor $v^{inicial}$.

Custo de retalhos

O custo de um retalho em estoque é multiplicado por 0,9 a cada período, ou seja, em 4.11, $\gamma = 0,9$. A escolha arbitrária feita de 0,9 é decorrente de vários testes computacionais realizados.

Exemplo

Exemplificamos a simulação para o problema considerando 12 períodos e a formação do estoque, demanda e limites para perda e retalho como descrito anteriormente. As

soluções são obtidas pelas heurísticas RAG - versão 1 (de acordo com Poldi e Arenales 2009, a heurística RAG - versão 1 apresentou melhores resultado para o problema de corte), RAG_A - versão 2 (a heurística RAG_A - versão 2 apresentou bons resultados (equivalentes aos resultados das heurísticas RAG_A - versões 1 e 3) para o problema de corte com sobras aproveitáveis) e RAG_A^P . Para o problema inicial (1º período), temos:

Tabela 6.32: Descrição das classes

| Objeto | Comprimento | Estoque |
|--------|--------------|--------------|
| 1 | $L_1 = 1000$ | $e_1 = 1000$ |
| 2 | $L_2 = 1100$ | $e_2 = 1000$ |

A demanda gerada a partir da escolha de itens do vetor $v^{inicial}$ que considera $v_1 = 0,01$ e $v_2 = 0,25$, é apresentada na Tabela 6.33. Para este exemplo, $m = 11$ foi gerado aleatoriamente, desta forma, 21 tipos de itens devem ser produzidos.

Tabela 6.33: Dados da demanda

| Item | Comprimento | demanda |
|------|----------------|----------------|
| 1 | $l_1 = 73$ | $d_1 = 468$ |
| 2 | $l_2 = 200$ | $d_2 = 254$ |
| 3 | $l_3 = 189$ | $d_3 = 345$ |
| 4 | $l_4 = 227$ | $d_4 = 487$ |
| 5 | $l_5 = 201$ | $d_5 = 300$ |
| 6 | $l_6 = 203$ | $d_6 = 334$ |
| 7 | $l_7 = 174$ | $d_7 = 415$ |
| 8 | $l_8 = 216$ | $d_8 = 322$ |
| 9 | $l_9 = 20$ | $d_9 = 442$ |
| 10 | $l_{10} = 26$ | $d_{10} = 264$ |
| 11 | $l_{11} = 233$ | $d_{11} = 6$ |
| 12 | $l_{12} = 242$ | $d_{12} = 4$ |
| 13 | $l_{13} = 166$ | $d_{13} = 4$ |
| 14 | $l_{14} = 228$ | $d_{14} = 9$ |
| 15 | $l_{15} = 32$ | $d_{15} = 3$ |
| 16 | $l_{16} = 225$ | $d_{16} = 1$ |
| 17 | $l_{17} = 232$ | $d_{17} = 1$ |
| 18 | $l_{18} = 107$ | $d_{18} = 5$ |
| 19 | $l_{19} = 202$ | $d_{19} = 8$ |
| 20 | $l_{20} = 61$ | $d_{20} = 5$ |
| 21 | $l_{21} = 75$ | $d_{21} = 7$ |

Como todos os itens gerados são pequenos, o comprimento máximo para a perda pequena em um objeto padronizado é θL_k com $\theta = 0,005$ e para um objeto não padronizado é $\beta = 5$. Como comparamos as soluções da heurística RAG_A^P com as soluções

das heurísticas RAG e RAG_A consideramos o valor $\beta = 5$ para todas as heurísticas (na Seção 6.1, β era uma pequena porcentagem de perda). O comprimento aceitável para um retalho é dado pela média do comprimento dos 10 primeiros itens demandados do vetor $v^{inicial}$, ou seja, $\delta = 152$.

Os custos dos objetos são calculados conforme 4.11. Para as heurísticas RAG e RAG_A , os retalhos gerados não têm descontos, ou seja, $c_k = cL_k, k = \bar{k} + 1, \dots, K$. Para a heurística RAG_A^P o custo dos objetos não padronizados é multiplicado por 0,9 a cada período, até serem utilizados.

Algumas informações de interesse das soluções das heurísticas residuais RAG, RAG_A e RAG_A^P para o primeiro e segundo períodos seguem nas Tabelas 6.34 e 6.38. Nestas tabelas, utilizamos (Comp. Perda) e (Comp. Retalho) para representar o comprimento das perdas e dos retalhos gerados, respectivamente, (Retalho Gerado) representa a quantidade de retalhos gerados, (Retalho Usado) representa a quantidade de objetos não padronizados (retalhos) utilizados e (Comp. Padronizado) representa o comprimento cortado de objetos padronizados.

Tabela 6.34: Solução para o 1º período

| | RAG | RAG_A | RAG_A^P |
|-------------------|--------|---------|-----------|
| Comp. Perda | 25 | 2 | 1 |
| Comp. Retalho | 773 | 796 | 897 |
| Retalho Gerado | 1 | 1 | 2 |
| Retalho Usado | 0 | 0 | 0 |
| Comp. Padronizado | 555700 | 555700 | 555800 |

Após obter a solução para o 1º período, temos os seguintes dados do estoque para o segundo período:

Tabela 6.35: Dados do estoque - Heurística RAG

| Objeto | Comprimento | Estoque |
|--------|--------------|--------------|
| 1 | $L_1 = 1000$ | $e_1 = 9925$ |
| 2 | $L_2 = 1100$ | $e_2 = 9563$ |
| 3 | $L_3 = 773$ | $e_3 = 1$ |

Tabela 6.36: Dados do estoque - Heurística RAG_A

| Objeto | Comprimento | Estoque |
|--------|--------------|--------------|
| 1 | $L_1 = 1000$ | $e_1 = 9925$ |
| 2 | $L_2 = 1100$ | $e_2 = 9563$ |
| 3 | $L_3 = 796$ | $e_3 = 1$ |

Tabela 6.37: Dados do estoque - Heurística RAG_A^P

| Objeto | Comprimento | Estoque |
|--------|--------------|--------------|
| 1 | $L_1 = 1000$ | $e_1 = 9926$ |
| 2 | $L_2 = 1100$ | $e_2 = 9562$ |
| 3 | $L_3 = 271$ | $e_3 = 1$ |
| 4 | $L_4 = 626$ | $e_4 = 1$ |

Para o 2º período a demanda é composta por 20 tipos de itens. As informações sobre as soluções das heurísticas RAG , RAG_A e RAG_A^P para o 2º período seguem na Tabela 6.38.

Tabela 6.38: Solução para o 2º período

| | RAG | RAG_A | RAG_A^P |
|-------------------|--------|---------|-----------|
| Comp. Perda | 0 | 5 | 2 |
| Comp. Retalho | 630 | 648 | 652 |
| Retalho Gerado | 1 | 1 | 1 |
| Retalho Usado | 1 | 1 | 2 |
| Comp. Padronizado | 569500 | 569500 | 569400 |

A partir dos dados fornecidos na Tabela 6.38 atualizamos o estoque de objetos para o 3º período e geramos uma nova demanda que neste caso é composta por 31 tipos de itens. De modo similar, este processo é repetido nos períodos restantes. A Figura 6.1 nos mostra a evolução das perdas e dos retalhos para este exemplo. Após 12 períodos, a simulação efetuada nos conduz aos dados da Tabela 6.39. Nesta tabela, (Comp. Perda) e (Comp. Retalho) representam o comprimento total das perdas e dos retalhos gerados, respectivamente, (Retalho Final) representa o comprimento final dos objetos não padronizados no final do 12º período, (Comp. Ret. Usado) representa o comprimento total

utilizado dos objetos não padronizados (retalhos) e (Comp. Padronizado) representa o comprimento total utilizado dos objetos padronizados.

Tabela 6.39: Solução depois de 12 períodos

| | RAG | RAG_A | RAG_A^P |
|-------------------|---------|---------|-----------|
| Comp. Perda | 122 | 25 | 4 |
| Comp. Retalho | 5990 | 7073 | 8349 |
| Retalho Final | 430 | 1627 | 248 |
| Comp. Ret. Usado | 5560 | 5446 | 8101 |
| Comp. Padronizado | 6507200 | 6508300 | 6506900 |

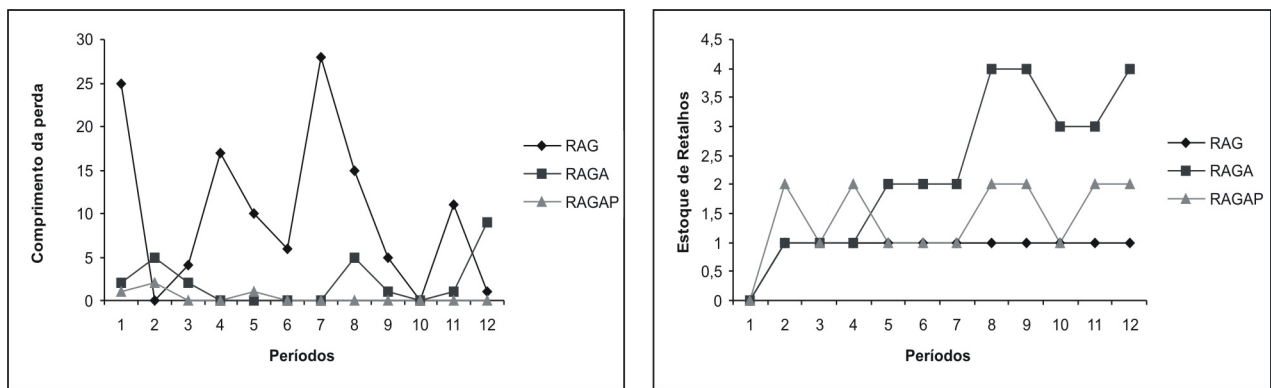


Figura 6.1: Comprimento da perda e número de retalhos por período.

Como observamos na primeira linha da Tabela 6.39 é conveniente trabalhar com soluções que procuram concentrar as sobras em poucos padrões de corte (retalho) e esperar por demandas futuras na tentativa de obter soluções com perdas baixas. Entretanto, devemos prestar atenção na quantidade de retalhos gerados pois, se não houver um controle sobre eles, o estoque pode aumentar muito e conseqüentemente, estes retalhos podem acabar sendo descartados como perdas. Além disso, a quantidade de objetos padronizados utilizados pode ser muito grande. Neste exemplo, podemos observar que a heurística RAG_A^P apresentou uma perda muito baixa quando consideramos uma simulação com 12 períodos, entretanto a heurística RAG manipula uma menor quantidade de retalhos durante os períodos (Figura 6.1).

A seguir, apresentamos os testes computacionais realizados considerando a simulação proposta para várias classes de exemplos.

6.2.1 Resultado de exemplares gerados aleatoriamente

Para a simulação, os testes realizados consideram problemas com itens de tamanhos pequenos, médios e variados. Para problemas envolvendo itens de tamanhos pequenos, atribuímos os valores 0,005 e 5 para os parâmetros θ e β , respectivamente, e para os problemas envolvendo itens de tamanhos médios e variados atribuímos os valores 0,01 e 10 para os parâmetros θ e β , respectivamente. Esta alteração nos valores de θ e β foi realizada devido ao fato de itens de tamanhos médios e variados incluírem itens maiores que, normalmente, são mais difíceis de serem combinados em um padrão de corte. O tamanho mínimo do retalho foi dado pela média do comprimento dos 10 primeiros itens do vetor $v^{inicial}$. Como comparamos os resultados da heurística RAG_A^P com as heurísticas RAG e RAG_A , também utilizamos o valor de $\beta = 5$ ou 10 para estas heurísticas.

Nas tabelas 6.40, 6.41 e 6.42 a seguir exibimos algumas quantidades de interesse considerando todos os períodos e todos os exemplares. Em cada linha destas tabelas, os maiores e menores valores obtidos estão em negrito e em itálico respectivamente. O cálculo da média é realizado considerando 20 simulações. Nestas tabelas, temos:

- Perda: comprimento médio da perda;
- Perda Acumulada: perda total acumulada no final do 12 períodos;
- Qt. Retalhos: quantidade média de retalhos em estoque ao final dos 12 períodos;
- Comp. Retalhos: comprimento médio dos retalhos em estoque ao final dos 12 períodos;
- Comp. Padroniz.: comprimento médio dos objetos padronizados cortados;
- Porcent. Perda: porcentagem média da perda total;
- Tempo: tempo médio de execução para cada simulação (em segundos).

Itens Pequenos: $\theta = 0,005$, $\beta = 5$

Tabela 6.40: Solução média - Itens pequenos

| | RAG | RAG_A | RAG_A^P |
|-----------------|---------------|----------------|-----------|
| Perda | 278.6 | 21.4 | 18.9 |
| Perda Acumulada | 5572 | 427 | 377 |
| Qt. Retalhos | 1.0 | 2.4 | 1.1 |
| Comp. Retalhos | 541.1 | 828.3 | 515.8 |
| Comp. Padroniz. | 5622726 | 5622755 | 5622440 |
| Porcent. Perda | 0.005% | 0.0004% | 0.0003% |
| Tempo | 38.4 | 41.6 | 37.8 |

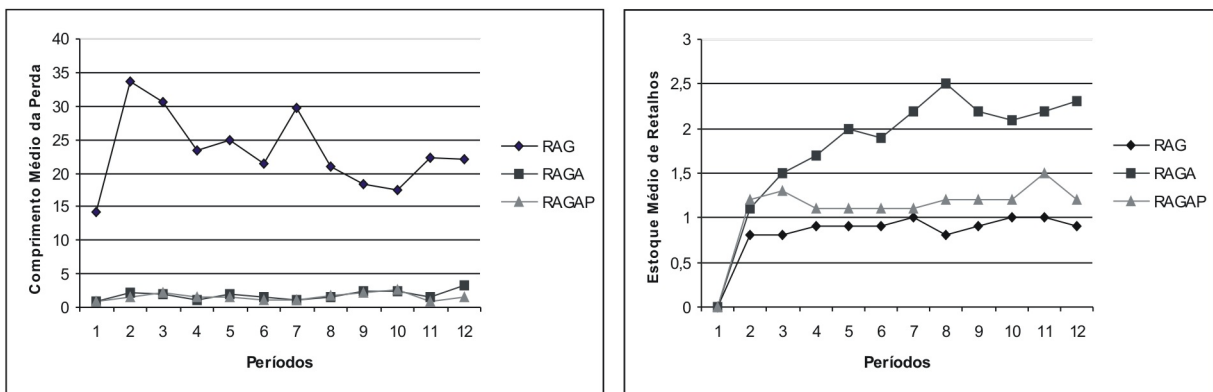


Figura 6.2: Perda e número médio de retalhos por período - Itens pequenos.

Para itens de tamanho pequenos, as soluções geradas pela heurística RAG apresentam uma perda maior que as perdas das heurísticas RAG_A e RAG_A^P (em média, dez vezes maiores). Do ponto de vista das perdas geradas, os resultados mostram claramente que é melhor manter retalhos em estoque e aguardar por futuras demandas. A heurística RAG apresenta, em média, o menor número de retalhos em estoque por período, enquanto que a heurística RAG_A^P apresenta uma quantidade ligeiramente maior. Considerando somente estas duas características, podemos dizer que a heurística RAG_A^P apresenta melhores resultados que as demais heurísticas quando temos itens pequenos para serem produzidos.

Itens Médios: $\theta = 0,01, \beta = 10$

Tabela 6.41: Solução média - Itens médios

| | RAG | RAG_A | RAG_A^P |
|-----------------|---------------|----------------|-----------|
| Perda | 2162.8 | 1534.2 | 1545.1 |
| Perda Acumulado | 42251 | 30595 | 30901 |
| Qt. Retalhos | 0.8 | 6.9 | 1.2 |
| Comp. Retalhos | 355.9 | 2559.5 | 483.6 |
| Comp. Padroniz. | 8609075 | 8610665 | 8608585 |
| Porcent. Perda | 0.03% | 0.02% | 0.02% |
| Tempo | 40.3 | 48.0 | 36.5 |

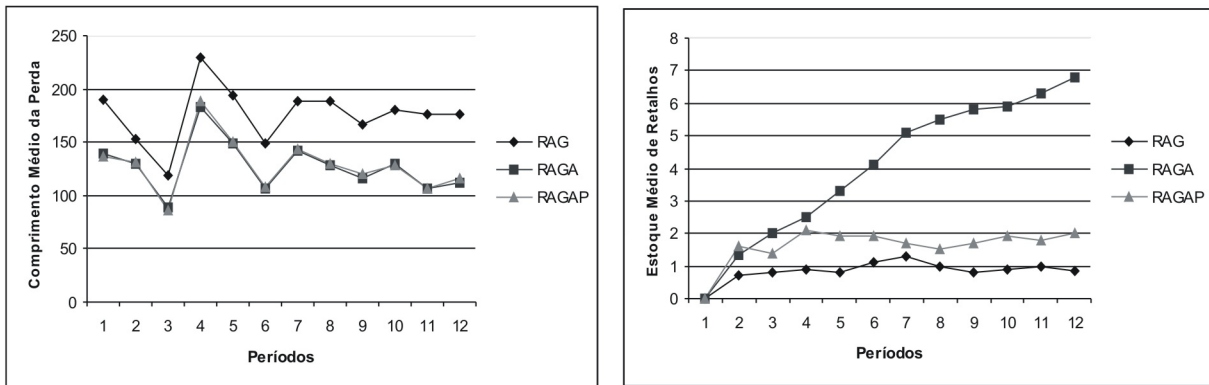


Figura 6.3: Perda e número médio de retalhos por período - itens médios.

Para itens médios, a perda gerada pela heurística RAG foi aproximadamente 40% maior que a perda das heurísticas RAG_A e RAG_A^P . Embora ainda seja vantajoso manter retalhos em estoque e esperar por demandas futuras, esta vantagem não é tão grande como quando consideramos itens pequenos. Isto ocorre devido a dificuldade de alocar itens maiores nos retalhos. Além disso, se não houver prioridade no corte dos retalhos, o estoque destes objetos pode aumentar muito que é uma situação inaceitável na prática.

Itens Variados: $\theta = 0,01, \beta = 10$

Tabela 6.42: Solução média - Itens variados

| | RAG | RAG_A | RAG_A^P |
|-----------------|----------------|--------------|-----------|
| Perda | 438.8 | 66.1 | 53.9 |
| Perda Acumulado | 8776 | 1321 | 1078 |
| Qt. Retalhos | 0.9 | 2.4 | 1.0 |
| Comp. Retalhos | 515.9 | 883.7 | 161.8 |
| Comp. Padroniz. | 7193870 | 7193865 | 7193450 |
| Porcent. Perda | 0.006% | 0.001% | 0.001% |
| Tempo | 38.1 | 42.6 | 35.3 |

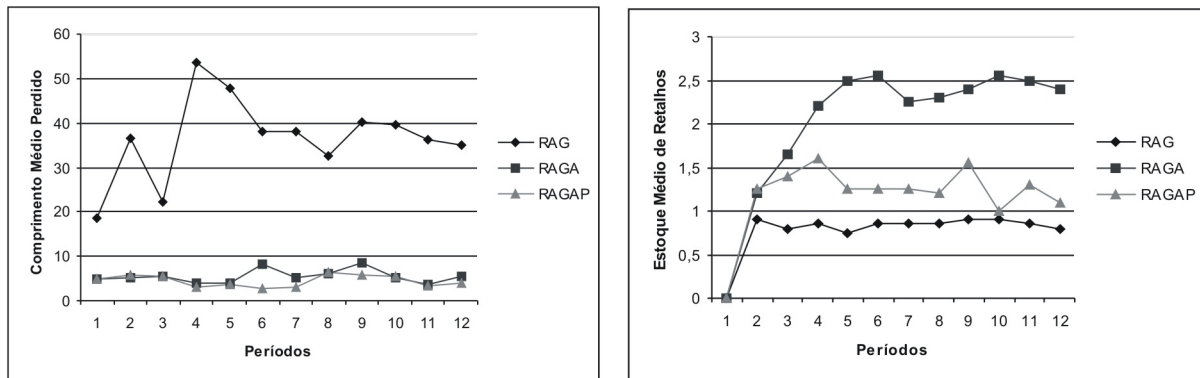


Figura 6.4: Perda e número médio de retalhos por período - itens variados.

Note que quando temos itens de tamanhos variados o comportamento das curvas é similar ao caso de itens de tamanhos pequenos. Desta forma, podemos concluir com os experimentos realizados que, para um melhor aproveitamento dos retalhos em estoque é importante que itens de tamanhos pequenos sejam demandados, isto também tem como consequência redução no comprimento das perdas geradas.

Com relação ao tempo computacional de execução das heurísticas, podemos observar pelas tabelas 6.40, 6.41 e 6.42 que todas elas demandam esforços bastante razoáveis para o uso na prática.

6.2.2 Resultados de exemplares da literatura

Nesta seção, apresentamos testes computacionais realizados com a heurística RAG_A^P utilizando os 320 exemplares (16 classes) gerados aleatoriamente (Seção 6.1.2), juntamente com as soluções obtidas pelo algoritmo COLA, desenvolvido por Gradisar *et al.* (1997). Este algoritmo é utilizado em nossa análise pelo fato de ordenar os objetos em estoque de modo não decrescente de comprimentos e, para cada objeto nesta seqüência, constrói um padrão de corte, ou seja, é um algoritmo que prioriza o uso de objetos não padronizados. Para esta análise, apresentamos novamente as soluções da heurística RAG - versão 1 (Seção 4.2.2), que fornece uma boa solução para o problema de corte quando o objetivo é a minimização das perdas e não há considerações para o aproveitamento de sobras e também apresentamos as soluções das heurísticas RAG_A - versões 1, 2 e 3 que geram boas soluções para o problema de corte de estoque com sobras aproveitáveis.

Na Tabela 6.43 apresentamos os valores médios obtidos para para os 320 exemplares, sendo que os maiores e menores valores estão obtidos estão em **negrito** e *itálico*, respectivamente. Nesta tabela, temos:

- Perda: comprimento médio da perda gerada;
- Comp. Ret.: comprimento total médio dos retalhos gerados;
- Num. Ret.: número médio de retalhos gerados;
- Comp. Não Pad.: comprimento médio dos objetos não padronizados cortados;
- Num. Não Pad.: número médio de objetos não padronizados utilizados;
- Comp. Pad.: comprimento médio dos objetos padronizados cortados;
- Tempo: tempo médio para executar os 320 exemplares (em segundos);
- Comp. Ret. Rem.: comprimento total de retalhos em estoque que não foram cortados (estoque remanescente);
- Num. Ret. Rem.: número total de retalhos em estoque que não foram cortados (estoque remanescente);
- Comp. Ret. Gerado: comprimento total de novos retalhos gerados;
- Num. Ret. Gerado: número total de novos retalhos gerados;

Tabela 6.43: Resultado computacional para os 320 exemplares

| | RAG_A^P | $RAG\ 1$ | $RAG_A\ 1$ | $RAG_A\ 2$ | $RAG_A\ 3$ | COLA |
|-------------------|---------------|-----------------|------------|-------------|----------------|---------------|
| Perda | 14,9 | 60,1 | 11,5 | 11,8 | 12,5 | 126,3 |
| Comp. Ret. | 835,9 | 587,4 | 552,8 | 601,9 | 34,9 | 685,8 |
| Num. Ret. | 2,1 | 2,6 | 2,6 | 2,8 | 0,3 | 3,4 |
| Comp. Não Pad. | 6971,3 | 1025,4 | 1041,6 | 1052,5 | 841,3 | 7065,3 |
| Num. Não Pad. | 27,7 | 4,3 | 4,4 | 4,5 | 3,7 | 28,1 |
| Comp. Pad. | 105197,2 | 110932,2 | 110881,9 | 110926,9 | 110612,5 | 105120,9 |
| Tempo | 39,52 | 22,5 | 22,7 | 81,8 | 22,4 | 44,85 |
| Comp. Ret. Rem. | 30072 | 1932760 | 1927576 | 1924088 | 1991672 | 0 |
| Num. Ret. Rem. | 115 | 7603 | 7571 | 7539 | 7795 | 0 |
| Comp. Ret. Gerado | 267488 | 187968 | 176896 | 192608 | 11168 | 219456 |
| Num. Ret. Gerado | 672 | 832 | 832 | 896 | 96 | 1088 |

Pela Tabela 6.43 observamos que a perda gerada pela heurística RAG_A - versão 1 é menor a perda das demais heurísticas e do algoritmo COLA. Entretanto a perda não deve ser considerada como único critério para avaliação das soluções, o número de retalhos em estoque também deve ser considerado simultaneamente para a análise.

Com relação ao estoque de retalhos, a heurística RAG_A^P tem 115 retalhos remanescentes em estoque e gera 672 novos, ou seja, ao final do processo de corte esta heurística fica com um estoque de 787 retalhos, enquanto que o algoritmo COLA utiliza todos os retalhos do estoque, porém, gera 1088 novos que ficarão em estoque. Com relação às demais heurísticas, temos que o estoque remanescente de retalhos é muito elevado e, além disso, uma grande quantidade de novos retalhos para o estoque é gerada.

Observe ainda que a heurística RAG_A^P e o algoritmo COLA utilizam menor comprimento de objetos padronizados para atender a demanda. Entretanto, pelas soluções apresentadas, temos que a heurística RAG_A^P é superior as demais heurísticas e ao algoritmo COLA. Embora a heurística RAG_A^P gere uma perda ligeiramente maior que as perdas das demais heurísticas de aproveitamento que não priorizam o corte de retalhos, ao final do processo de corte tem um estoque de retalhos muito baixo comparado aos estoques das demais heurísticas e do algoritmo COLA, que utiliza menor comprimento de objetos padronizados para atender a demanda e usa todos os retalhos do estoque, porém, apresenta uma perda elevada e gera um grande número de novos retalhos.

Durante o processo de corte em uma empresa, acreditamos que o decisor, por

razões operacionais e minimização de custos, optaria por um plano de corte com perdas menores e poucos retalhos em estoque. Neste caso, o procedimento que produz os melhores resultados com estas características é a heurística RAG_A^P .

Na Seção 6.3 apresentamos os testes computacionais realizados para avaliarmos o desempenho dos procedimentos desenvolvidos para o problema de corte bidimensional com aproveitamento de sobras.

6.3 Problema de corte bidimensional com sobras aproveitáveis

Para analisar o desempenho dos procedimentos heurísticos desenvolvidos para resolver o problema de corte bidimensional com sobras aproveitáveis, utilizamos dados reais de uma indústria de esquadrias metálicas de pequeno porte. As soluções foram obtidas pelos procedimentos heurísticos $Guloso_A^{2D}$ e RAG_A^{2D} - versão 2 (escolhemos esta versão pelo fato da heurística RAG - versão 2 selecionar os padrões de corte a serem utilizados de acordo com a menor perda gerada) e pelos seus respectivos procedimentos heurísticos clássicos, que não consideram o aproveitamento de sobras. Embora encontramos na internet softwares livres que consideram o aproveitamento de sobras para problemas de corte bidimensionais, não conseguimos utilizá-los para realizar uma comparação com procedimentos desenvolvidos pelo fato das versões serem muito limitadas.

Foram analisados 12 problemas, cada qual considerado como um período. Desta forma, a cada período resolvemos um problema de corte de estoque, sendo que as informações de estoque (objetos padronizados restantes e retalhos que foram gerados) para o problema no período atual são transferidas do período anterior. A cada período, uma nova demanda de itens deve ser atendida, que apesar de ser conhecida a priori é tratada como desconhecida no período anterior (a antecipação da produção de itens não é permitida).

Nestes problemas, o número de itens a serem produzidos varia entre 2 a 8 tipos, com uma demanda entre 2 a 468 itens de cada tipo. O número de placas padronizadas a serem cortadas varia entre 1 a 2 tipos.

Os valores atribuídos aos parâmetros utilizados pelo algoritmo Grafo E/OU_A para classificar as sobras geradas durante o processo de corte foram: $\psi_k = 0,05, k = 1, \dots, P$ (o valor 0,05 é utilizado quando o valor de ψ_k obtido durante o processo de corte é nulo), $\rho = 0,1, \phi = 0,25$ e $(\bar{\ell}_s, \bar{w}_s) = (500, 300)$ ou $(\bar{w}_s, \bar{\ell}_s) = (500, 300)$. Estes valores foram definidos considerando dados dos problemas a serem resolvidos, entretanto, outros valores poderiam ser utilizados.

Para a heurística RAG_A^{2D} o custo dos objetos não padronizados grandes (objetos $k = \bar{k} + 1, \dots, P$) é multiplicado por $\mu = 0,95$ a cada período e para os objetos não padronizados pequenos (objetos $k = P + 1, \dots, K$) o custo é multiplicado por $\gamma = 0,9$ a cada período, até serem utilizados.

Na Tabela 6.44 a seguir exibimos algumas informações de interesse para a análise do desempenho dos procedimentos heurísticos desenvolvidos considerando os 12 períodos. Nesta tabela, (Perda Perc.) representa o percentual da perda total gerada pelos procedimentos heurísticos, (Retalho Perc.) representa o percentual da área total dos retalhos não utilizados ao final dos 12 períodos, (Número Retalho) é o número total de retalhos em estoque ao final dos 12 períodos, (Perc. Padronizado) representa o percentual da área total utilizada dos objetos padronizados e (Tempo) representa o tempo total (em segundos) utilizado pelos procedimentos heurísticos para resolver os 12 problemas. Em cada linha desta tabela, os maiores e menores valores obtidos estão em negrito e em itálico respectivamente. A Figura 6.5 também nos auxilia na análise do desempenho dos procedimentos heurísticos.

Tabela 6.44: Solução para 12 períodos

| | Construtiva | | Residual | |
|------------------|---------------|--|------------|---------------------------------------|
| | <i>Gulosa</i> | <i>Gulosa</i> _A ^{2D} | <i>RAG</i> | <i>RAG</i> _A ^{2D} |
| Perda Perc. | 5,5% | 5,8% | 4,8% | 4,9% |
| Retalho Perc. | 79,1% | 1,6% | 7,4% | <i>0,7%</i> |
| Número Retalho | 13 | <i>2</i> | 5 | <i>2</i> |
| Área Padronizado | 98% | 93% | 96% | <i>92%</i> |
| Tempo | <i>1,26</i> | 1,74 | <i>3,1</i> | 4,78 |

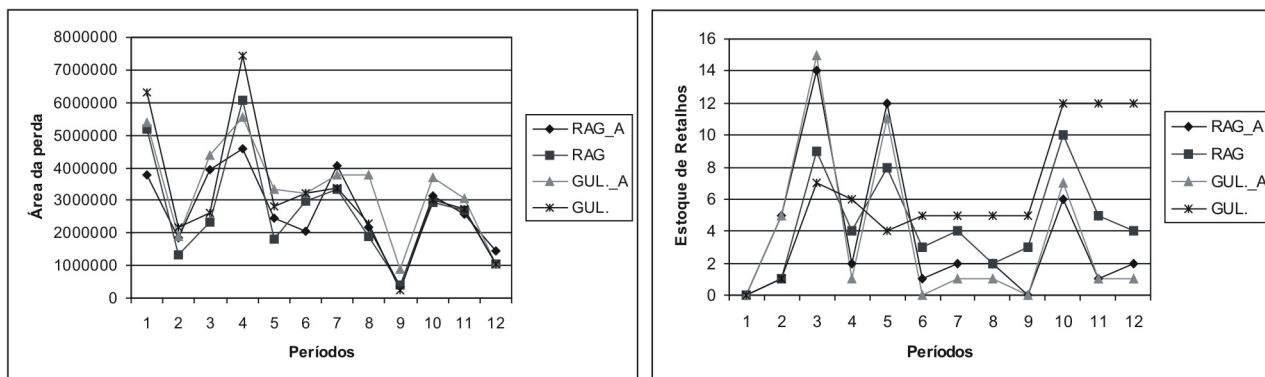


Figura 6.5: Perda e número total de retalhos por período.

Pela Tabela 6.44 observamos que a perda gerada pela heurística RAG é menor a perda das demais heurísticas. Entretanto, sua quantidade de retalhos em estoque ao final do processo de corte é maior que das heurísticas de aproveitamento. Observe que a heurística RAG_A^{2D} gera uma perda ligeiramente maior que a heurística RAG, porém, seu estoque de retalhos é baixo quando comparado aos estoques das heurísticas RAG e Gulosa. Além disso, estas heurísticas manipulam a maior quantidade de retalhos em quase todos os períodos (Figura 6.5).

As heurísticas de aproveitamento geram perdas maiores que suas respectivas versões clássicas pelo fato priorizarmos o corte dos retalhos e por permitirmos no máximo um retalho artificial em cada padrão de corte que apresenta perdas indesejáveis. Sem esta restrição, as soluções poderiam apresentar perdas muito menores, porém a quantidade de retalhos gerados seria muito maior.

Com as soluções obtidas, podemos dizer que a heurística RAG_A^{2D} é superior às demais heurísticas, pois gera um plano de corte ao longo dos períodos com perda baixa e com poucos retalhos no estoque. Outra característica que observamos a favor da heurística RAG_A^{2D} é que ela utiliza a menor área de objetos padronizados para atender a demanda.

Assim como no caso unidimensional, para os problemas de corte bidimensionais com sobras aproveitáveis notamos que é melhor a política de manter retalhos em estoque e aguardar por futuras demandas.

Obviamente, outras soluções poderiam ser encontradas alterando a ordem em que

os problemas foram resolvidos. Além disso, realizando algumas alterações no parâmetro ϕ ($\phi(L_k, W_k)$ estabelece as dimensões para que uma sobras seja classificada como retalho) a qualidade das soluções poderia ser afetada. Por exemplo, se utilizarmos $\phi = 0,21$, alguns retalhos gerados não seriam utilizados devido às suas dimensões, enquanto que com $\phi = 0,28$, as soluções não atenderiam a demanda devido ao tamanho aceitável para os retalhos. Neste caso, para que as demandas fossem atendidas, os retalhos gerados em alguns exemplares teriam que ser utilizados no mesmo período. Situações como essa, geralmente ocorrem quando a disponibilidade de objetos em estoque é baixa.

Com os testes computacionais realizados neste capítulo, observamos que a análise para a classificação do melhor procedimento heurístico não é trivial, pois as soluções possuem características importantes e conflitantes (no caso unidimensional, número de padrões com perda pequena, perda não tão pequena, retalho, comprimento da perda, entre outros). Desta forma, no Capítulo 7 a seguir, apresentamos uma metodologia mais simplificada para analisar o desempenho dos procedimentos heurísticos, que consiste em utilizar algumas técnicas da lógica *fuzzy* que facilitam a análise, comparação e classificação das soluções obtidas pelas heurísticas propostas.

Capítulo 7

Abordagem *Fuzzy* para o problema de corte com sobras aproveitáveis

Neste capítulo, apresentamos uma abordagem *fuzzy* para classificar as soluções dos problemas de corte unidimensionais com sobras aproveitáveis. Para a abordagem proposta, utilizamos o problema definido na Seção 3.1 e alguns procedimentos heurísticos descritos na Seção 4.

7.1 Definição dos parâmetros

Para classificar as soluções geradas para o problema de corte com sobras aproveitáveis em *ideal*, *aceitável* e *indesejável* (Seção 3.1), alguns parâmetros e critérios foram definidos, sendo que o número de objetos cortados com perda pequena, não tão pequena e retalhos, influenciam diretamente nesta classificação. Além disso, os critérios estabelecidos nem sempre são satisfeitos simultaneamente (dificultando a análise), pois na maioria das vezes existem soluções com perdas pequenas e retalhos distribuídos em muitos objetos cortados ou, soluções com perdas não tão pequenas e retalhos concentrados em poucos objetos, entre outras.

Neste capítulo, devido à facilidade em processar e analisar o desempenho dos procedimentos heurísticos utilizando as técnicas da lógica *fuzzy*, redefinimos a maneira de classificar as soluções (Definição 3.1), com a finalidade de tornar a análise mais precisa. Os dados utilizados anteriormente para análise do problema (números de objetos cortados

com perda pequena, não tão pequena e retalho) também foram alterados e passamos a utilizar para classificação dos procedimentos heurísticos a perda relativa nos objetos padronizados, não padronizados e o retalho líquido relativo, ou seja, analisamos o percentual de perda com relação ao comprimento cortado de objetos padronizados e não padronizados e o percentual de retalhos que compõem o novo estoque (estoque de objetos não padronizados após o processo de corte) em relação ao estoque de retalhos inicial.

Definição 7.1 *Para o problema de corte de estoque com sobras de material aproveitáveis, as soluções são definidas como:*

- **Solução ideal:** *quando a redução de retalhos (objetos não padronizados) for bem significativa e a perda relativa nos objetos padronizados e não padronizados for muito pequena;*
- **Solução desejável:** *quando a redução de retalhos for bem significativa, a perda relativa nos objetos padronizados for muito pequena e a perda relativa nos objetos não padronizados for pequena;*
- **Solução aceitável:** *quando a redução de retalhos for pouco significativa e a perda nos objetos padronizados for pequena;*
- **Solução indesejável:** *quando o estoque de retalhos se mantém estável e a perda relativa nos objetos padronizados for pequena;*
- **Solução inaceitável:** *quando o estoque de retalhos for ampliado.*

Como podemos observar, esta nova classificação requer a caracterização das quantidades dos termos *muito pequena* e *pequena*, *redução bem significativa*, *pouco significativa*, *estável* ou *ampliado*.

Para a quantificação dos termos *muito pequena* e *pequena*, utilizamos parâmetros arbitrários que podem ser definidos pelo usuário (decisor). O usuário também pode utilizar os seguintes parâmetros:

- *muito pequena:* se perda relativa $< \xi_1^p$ (ξ_1^{np});
- *pequena:* se perda relativa $< \xi_2^p$ (ξ_2^{np}).

sendo que $0 < \xi_1^p < \xi_2^p < 1$ é o limite da perda relativa nos objetos padronizados e $0 < \xi_1^{np} < \xi_2^{np} < 1$ é limite da perda relativa nos objetos não padronizados.

A perda relativa é calculada pela razão entre a perda total e o comprimento dos objetos cortados, ou seja,

$$\text{Perda relativa} = \frac{\text{comprimento total perdido}}{\text{comprimento total de objetos cortados}}$$

em que o comprimento perdido e o comprimento total dos objetos cortados devem ser referente ao mesmo tipo de objeto (padronizado ou não padronizado).

Para obter as variações do estoque de objetos não padronizados (retalhos), calculamos o retalho líquido relativo a este estoque:

$$\text{Retalho líquido relativo} = \frac{\text{número de objetos retalho em estoque} - \text{retalhos utilizados} + \text{novos retalhos}}{\text{número de objetos retalho em estoque}}$$

no qual o número de objetos retalho em estoque é a quantidade de objetos não padronizados em estoque antes de iniciar o processo de corte.

Para a quantificação dos termos referentes ao retalho líquido relativo, definimos a redução do estoque da seguinte maneira:

- *bem significativa*: se retalho líquido relativo $\leq \omega_1$;
- *pouco significativa*: se retalho líquido relativo $\leq \omega_2$;
- *estável*: se retalho líquido relativo $\leq \omega_3$;
- *ampliado*: se retalho líquido relativo $> \omega_3$.

em que os parâmetros ω_1, ω_2 e ω_3 também são arbitrários e podem ser definidos pelo usuário de forma que $0 < \omega_1 < \omega_2 < \omega_3 \leq 1$.

Convém ressaltar que o uso do retalho líquido na análise das soluções só deve ser utilizado quando existir uma quantidade razoável de objetos em estoque. Este critério não deve ser utilizado quando, por exemplo, o estoque de retalhos é nulo, pois se apenas um retalho for gerado durante o processo de corte, o retalho líquido relativo será infinito.

Embora vários procedimentos heurísticos tenham sido desenvolvidos para resolver o problema de corte de estoque com sobras aproveitáveis (Seção 4), comparamos apenas

as soluções das heurísticas construtivas FFD (*First Fit Decreasing*) e FFD_A, residuais RAG - versão 1 e RAG_A - versão 1 e do algoritmo COLA, desenvolvido por Gradisar *et al.* (1997). Ressaltamos que outros procedimentos heurísticos podem ser utilizados para mostrar a facilidade de análise das soluções a partir da lógica *fuzzy*.

7.2 Técnicas de inferência *fuzzy*

A lógica *fuzzy* é baseada na teoria dos conjuntos *fuzzy*. Esta é uma generalização da teoria dos conjuntos tradicionais para resolver os paradoxos gerados à partir da classificação “verdadeiro ou falso” da lógica clássica. Tradicionalmente, uma proposição lógica tem dois extremos: ou “completamente verdadeira” ou “completamente falsa”. Entretanto, na lógica *fuzzy*, uma premissa varia em grau de verdade de 0 a 1, tornando-se parcialmente verdadeira ou parcialmente falsa.

A força da Lógica *Fuzzy* deriva da sua habilidade em inferir conclusões e gerar respostas baseadas em informações vagas, ambíguas e qualitativamente incompletas e imprecisas. Neste aspecto, os sistemas de base *fuzzy* têm habilidade de raciocinar de forma semelhante à dos humanos. Seu comportamento é representado de maneira muito simples e natural, levando à construção de sistemas compreensíveis e de fácil manutenção.

Para Pedrycz (1998), o uso de conjuntos *fuzzy* produz uma base para um meio sistemático para a manipulação das concepções incertas e vagas. Em particular, os conjuntos *fuzzy* podem ser empregados para representar as variáveis lingüísticas. Uma variável lingüística pode ser considerada como qualquer variável cujo valor é um número *fuzzy* (os números *fuzzy* são constituídos de conjuntos *fuzzy*, definidos em universos de discurso discretos ou contínuos) ou cujos valores são definidos em termos lingüísticos. As principais operações entre variáveis lingüísticas são realizadas através da utilização dos conectivos “e” (operador de intersecção τ - norma), “ou” (operador de união S -norma) e “não”. Assim, dados dois termos A e B de uma determinada variável lingüística, as operações compostas por “ A e B ” e “ A ou B ”, são definidas em termos dos seus graus de pertinência como:

$$\mu_A(x) \text{ e } \mu_B(x) = \mu_A(x)\tau\mu_B(x)$$

$$\mu_A(x) \text{ ou } \mu_B(x) = \mu_A(x)S\mu_B(x)$$

em que $x \in U$ (Universo de discurso).

Utilizando para o operador τ - norma a função mínimo (min) e para o operador S -norma a função máximo (max), temos:

$$\mu_A(x) \text{ e } \mu_B(x) = \min(\mu_A(x), \mu_B(x))$$

$$\mu_A(x) \text{ ou } \mu_B(x) = \max(\mu_A(x), \mu_B(x))$$

Para a operação de complemento “não” temos a seguinte expressão:

$$\text{não}(\mu_A(x)) = 1 - \mu_A(x), \quad x \in U;$$

Se x e y forem variáveis lingüísticas compostas respectivamente por um conjunto de termos A_1, A_2, \dots, A_n e B_1, B_2, \dots, B_m o problema básico do processo de inferência é encontrar uma função de pertinência B' que represente a consequência da aplicação simultânea das regras da forma “se - então”.

Normalmente, os processos de inferência *fuzzy* aplicados em determinadas regras são baseados na regra de Modus Ponens generalizada que é explicitada para um conjunto observável A' como:

- Fato: x é A' ;
- Regra: se x é A então y é B ;
- Consequência: y é B' .

Logo, se o conjunto A implica diretamente no conjunto B ($A \rightarrow B$), então esta operação de implicação pode ser transformada em uma relação de implicação $R_{A \rightarrow B}(x, y)$. Desta forma, para obtermos o conjunto B' , basta compor o conjunto que denota um fato observável A' , com a relação de implicação $R_{A \rightarrow B}(x, y)$, utilizando a operação de composição max-min, (Pedrycz, 1998), ou seja:

$$B' = A'(x) \circ R_{A \rightarrow B}(x, y).$$

7.2.1 Relações de implicação

A obtenção da função de pertinência relativa à relação de implicação $R_{A \rightarrow B}$ pode ser computada utilizando vários operadores de implicação (Pedrycz, 1998). Considerando as variáveis linguísticas A e B , a função de pertinência $\mu_{R_{A \rightarrow B}}(x, y)$ pode ser obtida, por exemplo, através do operador de Mandami, cuja idéia é descrever determinados processos por meio de variáveis linguísticas e usar estas variáveis como entrada para regras de controle. Formalmente, temos:

$$\mu_{R_{A \rightarrow B}}(x, y) = \min(\mu_A(x); \mu_B(y))$$

Definição 7.2 *Singleton é um caso particular de conjunto fuzzy normalizado, cujo suporte é um único ponto $x \in X$ com $\mu(x) = 1$.*

Os conjuntos *singleton* são especialmente utilizados para mapear as grandezas de entrada do sistema *fuzzy* que geralmente são representadas por valores pontuais.

De acordo com Wang (1996), genericamente, um sistema *fuzzy* é composto de quatro componentes conectados:

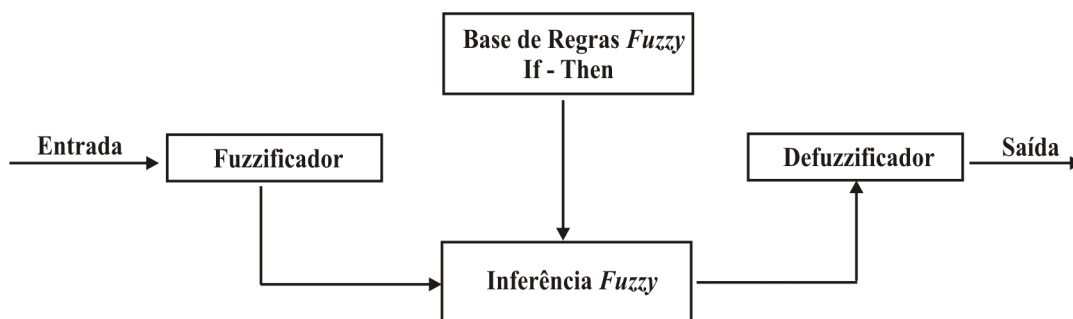


Figura 7.1: Sistema *Fuzzy*.

- **Fuzzificadores** - A principal função de um fuzzificador é converter os valores reais de entrada em um grau de pertinência a conjuntos *fuzzy*. Dentre todos os fuzzificadores existentes, um dos mais utilizados é o *singleton*.
- **Base de regras *fuzzy*** - A base de regras *fuzzy* consiste de um conjunto de regras linguísticas “IF - THEN” sendo considerada a parte fundamental de um sistema

fuzzy, uma vez que todos os outros componentes são utilizados para implementar as regras de modo eficiente e razoável.

- **Inferência *fuzzy*** - Em um processo de inferência *fuzzy*, os princípios da lógica *fuzzy* são usados para combinar as regras *fuzzy* “IF - THEN” existentes na base de regras com os dados *fuzzy* de entrada, de modo a inferir as ações de saída *fuzzy*. Assim, se considerarmos um sistema composto por duas entradas e uma saída, com as variáveis lingüísticas de entrada x e y compostas por um conjunto de termos *fuzzy* $\{A_1, A_2, \dots, A_n\}$, $\{B_1, B_2, \dots, B_m\}$ e uma variável de saída z especificada no conjunto de termos $\{C_1, C_2, \dots, C_p\}$, temos:

- Fato 1: x é A' ;
- Fato 2: y é B' ;
- Regra 1: se (x é A_1) e (y é B_1) então z é C_1 ;
- Regra 2: se (x é A_2) e (y é B_2) então z é C_2 ;
- Conseqüência: z é C' .

Para obtermos a relação de implicação $R_{AeB \rightarrow C}$, basta aplicar o conectivo lógico “e” em todas as regras ativadas, levando-se em consideração somente a relevância em termos do grau de ativação e, em seguida, detectar todas as regras ativadas e suas respectivas regiões *fuzzy* de saída. Através da operação de composição max-min, temos para cada regra ativada k a relação:

$$C'_k(z) = AB'(x, y) \circ R_{AB \rightarrow C}(x, y, z)$$

Combinando todas as regiões *fuzzy* de saída $C'_k(z)$, obtemos uma região *fuzzy* $C'(z)$ que representa a agregação (denotada por Ag) de todas as contribuições $C'_k(z)$, ou seja, $C'_k = Ag(C'_1, C'_2, \dots, C'_k)$, em que k é o número de regras ativadas.

A agregação é a forma de combinar os conseqüentes (conjuntos *fuzzy* de saída) gerados à partir das regras ativadas. Normalmente se faz agregação pelo máximo, ou seja, entre os diversos conseqüentes de um mesmo conjunto gerados na implicação, seleciona-se o de maior grau de pertinência.

- **Defuzzificadores** - O defuzzificador especifica um ponto na saída que melhor representa o conjunto *fuzzy*. Na escolha de um defuzzificador, os critérios de plausidade, simplicidade computacional e de continuidade devem ser considerados. Segundo Wang (1996) os defuzzificadores mais utilizados são: centro de área, média dos máximos e primeiro máximo.

7.2.2 Métodos de defuzzificação

Os métodos de defuzzificação são frequentemente baseados em idéias heurísticas. Entretanto, esses métodos também podem ser caracterizados por suas propriedades (matemáticas) formais. A defuzzificação não é somente relevante para controles *fuzzy*, mas também para outros tipos de problemas.

Para determinar a região *fuzzy* gerada por todas as regras ativadas, devemos aplicar um método de defuzzificação no conjunto $C'(z)$ resultante da agregação de todos os conjuntos *fuzzy* de saída $C'_k(z)$. Dentre os vários métodos definidos para o processo de defuzzificação, utilizamos o Centro de Área (CDA), cuja idéia é agregar as informações sobre possíveis ações de controle que são representadas pela função de pertinência. A solução é um termo comum que combina a contribuição de cada regra ativada das conseqüências. Formalmente, temos:

$$CDA = \frac{\sum_{k=1}^n \mu_{C'}(V_k) V_k}{\sum_{k=1}^n \mu_{C'}(V_k)}$$

em que n é o número de discretizações realizadas no universo de discurso de C' e V_k são os valores do universo de discurso de C' .

7.3 Experimentos computacionais

Nesta seção, para avaliar os procedimentos heurísticos descritos no Capítulo 4, apresentamos os resultados computacionais de exemplares gerados aleatoriamente conforme o gerador de problemas descrito na Seção 6.1.2. Para estes exemplares, 18 classes

de exemplos foram geradas considerando as seguintes modificações no gerador (Seção 6.1.2):

- *Comprimentos dos objetos padronizados*: $L_1 = 900$ e $L_2 = 1100$;
- *Comprimentos dos itens*: os itens de comprimento ℓ_i são gerados aleatoriamente no intervalo $[v_1L, v_2L]$, em que L é o valor médio de $L_k, k = 1, 2, v_1 = 0,001, v_2 = 0,25$ (para itens de comprimento pequenos - P) e $v_2 = 0,5$ (para itens de comprimento variados - V);
- *Demanda*: d_i é gerada aleatoriamente no intervalo $[1, 5]$;

Note que a demanda dos itens é baixa, o que reflete uma situação freqüente em indústrias de pequeno porte, cujas carteiras de pedidos são bem variadas.

Na tabela 7.1 estão descritas as 18 classes de exemplos geradas. Para estas classes, utilizamos $\theta = 0,005, \beta = 0,5$ e $\delta =$ média dos comprimentos dos itens demandados.

Tabela 7.1: Descrição das classes

| Classe | Parâmetros | | |
|--------|------------|-----|-------|
| | K | m | Itens |
| 1 | 5 | 10 | P |
| 2 | 5 | 10 | V |
| 3 | 5 | 20 | P |
| 4 | 5 | 20 | V |
| 5 | 5 | 40 | P |
| 6 | 5 | 40 | V |
| 7 | 7 | 10 | P |
| 8 | 7 | 10 | V |
| 9 | 7 | 20 | P |
| 10 | 7 | 20 | V |
| 11 | 7 | 40 | P |
| 12 | 7 | 40 | V |
| 13 | 9 | 10 | P |
| 14 | 9 | 10 | V |
| 15 | 9 | 20 | P |
| 16 | 9 | 20 | V |
| 17 | 9 | 40 | P |
| 18 | 9 | 40 | V |

Para cada classe descrita na Tabela 7.1, 50 exemplares foram gerados. A média dos resultados obtidos com testes computacionais realizados para os 900 exemplares (18 classes

× 50 exemplares) são apresentadas na Tabela 7.2 a seguir. Nesta tabela, (Média Prop) representa a média da perda relativa nos objetos padronizados, (Média Pronp) representa a média da perda relativa nos objetos não padronizados e (Média Rlr) representa o retalho líquido relativo.

Em cada linha da Tabela 7.2 os maiores e os menores valores estão em negrito e itálico, respectivamente. Os resultados médios de cada classe de exemplos geradas são apresentados nas Tabelas A.14, A.15 e A.16 do Apêndice A.

Tabela 7.2: Perda média relativa nos objetos padronizados, não padronizados e retalho líquido relativo

| | Construtiva | | | Residual | |
|--------------------|---------------|------------|------------------------|---------------|--------------------------|
| | <i>COLA</i> | <i>FFD</i> | <i>FFD_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> |
| Média Prop | 0,0183 | 0,0088 | <i>0,0002</i> | 0,0021 | 0,0005 |
| Média Pronp | 0,0556 | 0,0328 | 0,0090 | 0,0977 | <i>0,0070</i> |
| Média Rlr | <i>0,2540</i> | 0,6447 | 0,9540 | 0,9110 | 0,9508 |

Diferentemente dos procedimentos heurísticos desenvolvidos, o algoritmo COLA ordena os objetos de modo não decrescente e, para cada objeto nesta seqüência, constrói um padrão de corte que prioriza o uso de objetos não padronizados. Desta forma, o algoritmo COLA consegue reduzir o estoque de objetos não padronizados em, aproximadamente, 75%. As heurísticas propostas para o problema de aproveitamento reduzem gradativamente o estoque de objetos não padronizados, uma vez que o uso desses objetos é estimulado apenas por permitir que uma perda maior em relação aos objetos padronizados seja gerada.

Em relação às perdas, observamos que as heurísticas propostas apresentam soluções superiores ao algoritmo COLA e às heurísticas originais (clássicas). A redução das perdas deve-se ao fato de que sobras indesejáveis são alteradas para se tornarem uma perda pequena ou um retalho (objeto não padronizado futuro). Entretanto, tal redução pode originar um efeito indesejável para a solução, pois muitos retalhos podem ser gerados. O aumento de objetos não padronizados futuros pode comprometer a qualidade da solução, conforme a Definição 7.1, tornando uma solução *indesejável* ou até mesmo *inaceitável*.

7.4 Análise das soluções

A classificação das soluções obtidas (Seção 7.3) é realizada aplicando-se o processo de inferência *fuzzy* (Seção 7.2), visto que vários fatores são considerados simultaneamente na avaliação das soluções.

Para o processo de inferência *fuzzy*, utilizamos o Toolbox Fuzzy Logic do Software MatLab 7.1. Para a análise dos resultados, relacionamos os dados da Tabela 7.2. Estes dados foram escolhidos para a análise da solução final por serem fatores que influenciam diretamente na qualidade da solução, ou seja, o comprimento total perdido deve estar diretamente relacionado com a quantidade de novos retalhos que devem retornar ao estoque.

Para os parâmetros definidos na Seção 7.1, consideramos os valores 0,01 e 0,015 para os parâmetros ξ_2^p e ξ_1^p , respectivamente, e para os parâmetros ξ_1^{np} e ξ_2^{np} , definimos os valores 0,015 e 0,03, respectivamente. Considerando o retalho líquido relativo, definimos $\omega_1 = 0,5$, $\omega_2 = 0,8$ e $\omega_3 = 1,0$.

As variáveis de entrada para o sistema *fuzzy* proposto são: a perda relativa nos objetos padronizados e não padronizados e o retalho líquido relativo. A variável de saída é a classificação das soluções em *ideal* (ID), *desejável* (DE), *aceitável* (AC), *indesejável* (IND) e *inaceitável* (INA), conforme a Definição 7.1. As funções de pertinência relacionadas à esses dados têm como formato os padrões geométricos mostrados nas Figuras 7.2 e 7.3 com os parâmetros: MPE (Muito Pequenas), PE (Pequenas) e G (Grandes) para as perdas relativas e BS (Bem Significativa), PS (Pouco Significativa), E (Estável) e A (Ampliada) para os retalhos líquidos relativos. Na Figura 7.2, para todos os valores referentes à perda relativa nos objetos padronizados e não padronizados e retalho líquido relativo, adicionamos a média de seus respectivos desvio padrão, sendo que:

- Desvio padrão médio da perda nos objetos padronizados: 0,008;
- Desvio padrão médio da perda nos objetos não padronizados: 0,061;
- Desvio padrão médio do retalho líquido relativo: 0,166.

Desta forma, se considerarmos o primeiro valor na Figura 7.2 (b) temos que 0,666 foi obtido assumindo $\omega_1 = 0,5$ adicionado do desvio padrão 0,166.

Para construção dos padrões geométricos da Figura 7.2 nos baseamos nos valores da Tabela 7.2. Entretanto, para os padrões geométricos da Figura 7.2 (a e c) multiplicados os valores obtidos por 100, devido a grandeza que os mesmos apresentam.

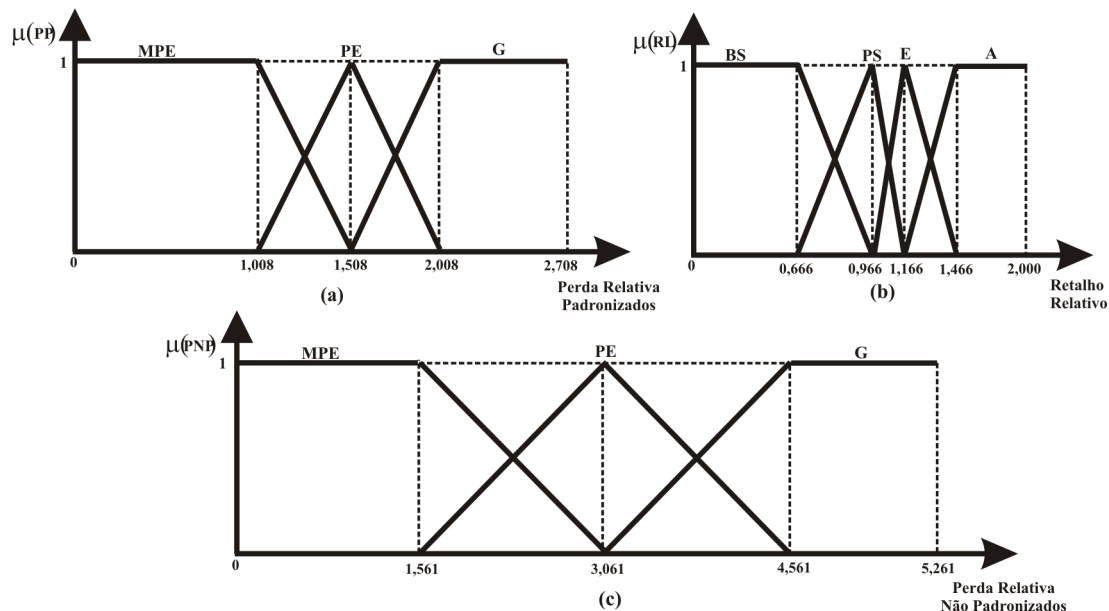


Figura 7.2: Padrões geométricos: (a) Perda Relativa nos Objetos Padronizados, (b) Retalho Líquido Relativo, (c) Perda Relativa nos Objetos não Padronizados.

Os padrões geométricos apresentados na Figura 7.3 são referentes aos dados de saída estabelecidos para o sistema *fuzzy*.

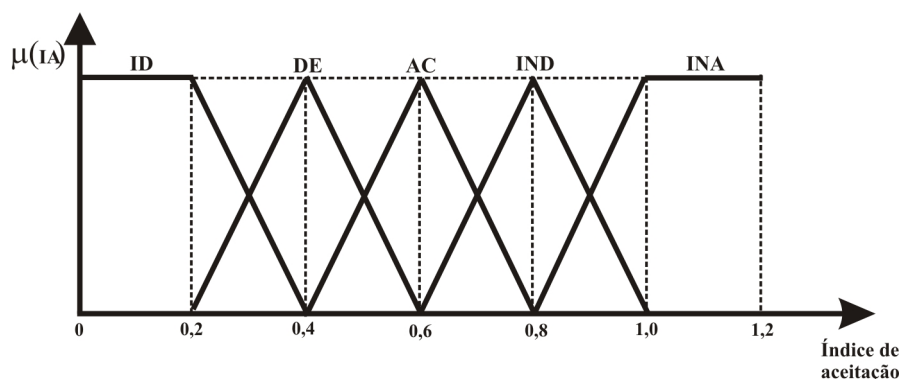


Figura 7.3: Saída *Fuzzy* - Índice de Aceitabilidade.

Observe que os padrões geométricos apresentados na Figura 7.3 são igualmente espaçados. Isto ocorre porque no problema de reconhecimento de padrões, todas as classes devem ter a mesma probabilidade de serem escolhidas, por isso é usual normalizar a saída de dados (entre 0-1 ou outro intervalo conveniente). Ressaltamos que o processo de reconhecimento de padrão é qualitativo, ou seja, o que importa é a classe a qual pertence uma dada solução.

O ajuste entre os padrões geométricos e a saída *fuzzy* é geralmente feito com base na fixação dos conseqüentes e na alteração dos antecedentes (funções de pertinência).

As regras do sistema *fuzzy* que representam o conhecimento base para análise das soluções do problema têm o seguinte formato:

Regra 1: Se (a perda relativa nos objetos padronizados for **muito pequena**) e (a perda relativa nos objetos não padronizados for **muito pequena**) e (a redução do estoque de retalhos for **bem significativa**), então a solução é *ideal*.

De maneira simplificada, a Tabela 7.3 a seguir sintetiza as 36 possíveis regras *fuzzy*. Nesta tabela, consideramos PerdaRelPad e PerdaRelNPad a perda relativa nos objetos padronizados e não padronizados, respectivamente. Bem Signif. e Pouco Signif. são referentes aos termos bem significativa e pouco significativa, respectivamente.

Tabela 7.3: Regras Fuzzy

| PerdaRelPad\PerdaRelNPad | Retalho Líquido Relativo | | | |
|---|--------------------------|--------------------|--------------------|--------------------|
| | Bem Signif. | Pouco Signif. | Estável | Ampliado |
| Muito Pequena \ Muito Pequena | <i>Ideal</i> | <i>Desejável</i> | <i>Aceitável</i> | <i>Aceitável</i> |
| Muito Pequena \ Pequena | <i>Desejável</i> | <i>Desejável</i> | <i>Aceitável</i> | <i>Indesejável</i> |
| Muito Pequena \ Grande | <i>Aceitável</i> | <i>Aceitável</i> | <i>Aceitável</i> | <i>Indesejável</i> |
| Pequena \ Muito Pequena | <i>Desejável</i> | <i>Aceitável</i> | <i>Aceitável</i> | <i>Indesejável</i> |
| Pequena \ Pequena | <i>Desejável</i> | <i>Aceitável</i> | <i>Aceitável</i> | <i>Indesejável</i> |
| Pequena \ Grande | <i>Aceitável</i> | <i>Aceitável</i> | <i>Indesejável</i> | <i>Indesejável</i> |
| Grande \ Muito Pequena | <i>Aceitável</i> | <i>Aceitável</i> | <i>Indesejável</i> | <i>Indesejável</i> |
| Grande \ Pequena | <i>Aceitável</i> | <i>Indesejável</i> | <i>Indesejável</i> | <i>Indesejável</i> |
| Grande \ Grande | <i>Indesejável</i> | <i>Indesejável</i> | <i>Indesejável</i> | <i>Inaceitável</i> |

Os operadores que utilizamos em todo processo de inferência são:

- *Implicação*: Mandami;

- *Agregação*: Máximo;
- *Defuzzificação*: Centro de Área (CDA);
- τ - *norma*: Mínimo.

Para uma melhor compreensão do desenvolvimento do sistema *fuzzy* utilizado para classificar as soluções do problema de corte com sobras aproveitáveis, ilustramos a seguir as etapas do processo desenvolvido considerando as 36 regras apresentadas na Tabela 7.3 e os dados de entrada da heurística RAG_A 1 (Tabela 7.2).

Inicialmente, localizamos as regras que foram ativadas a partir dos dados de entrada da heurística RAG_A 1 e dos padrões geométricos definidos (Figura 7.2). Para os valores fornecidos pela heurística RAG_A temos apenas duas regras ativadas:

Regra 1: Se (a perda relativa nos objetos padronizados for **muito pequena**) e (a perda relativa nos objetos não padronizados for **muito pequena**) e (a redução do estoque de retalhos for **bem significativa**), então a solução é *ideal*.

Regra 2: Se (a perda relativa nos objetos padronizados for **muito pequena**) e (a perda relativa nos objetos não padronizados for **muito pequena**) e (a redução do estoque de retalhos for **pouco significativa**), então a solução é *desejável*.

A representação geométrica das regras que foram ativadas são apresentadas na Figura 7.4. Observe que na representação geométrica do retalho líquido relativo duas regras foram ativadas (BS: bem significativa e PS: pouco significativa).

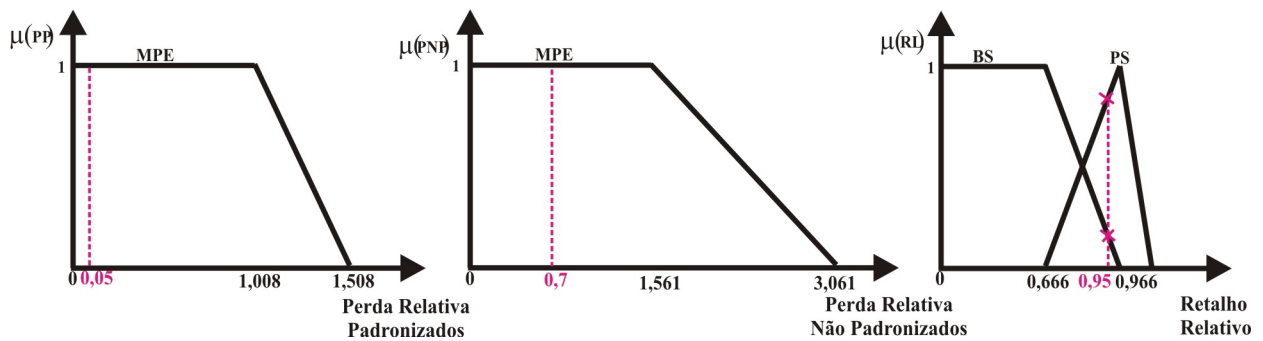


Figura 7.4: Ilustração gráfica do processo de Inferência *Fuzzy*.

O próximo passo consiste em desmembrar os termos das respectivas funções de pertinência envolvidas com as regras ativadas e verificar suas respectivas regiões *fuzzy* de saída. Estas regiões de saída são obtidas utilizando os padrões geométricos da Figura 7.3.

Regra 1

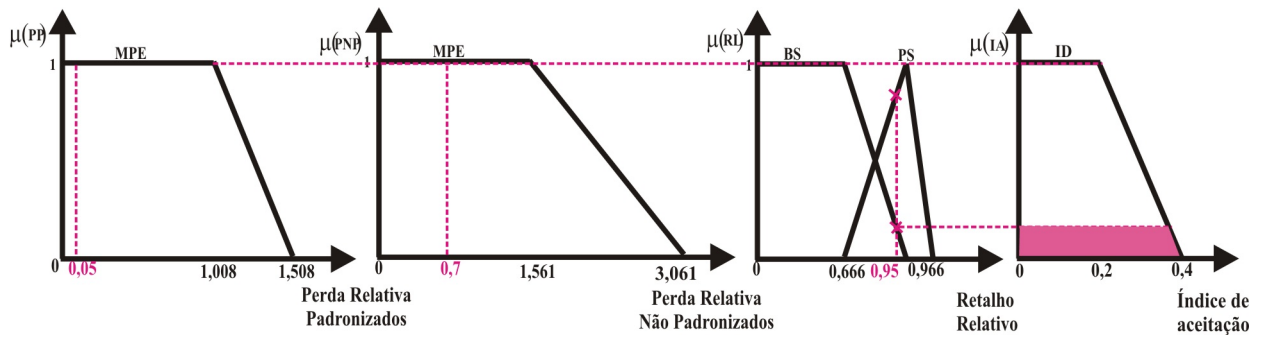


Figura 7.5: Região ativada: Regra 1.

Regra 2

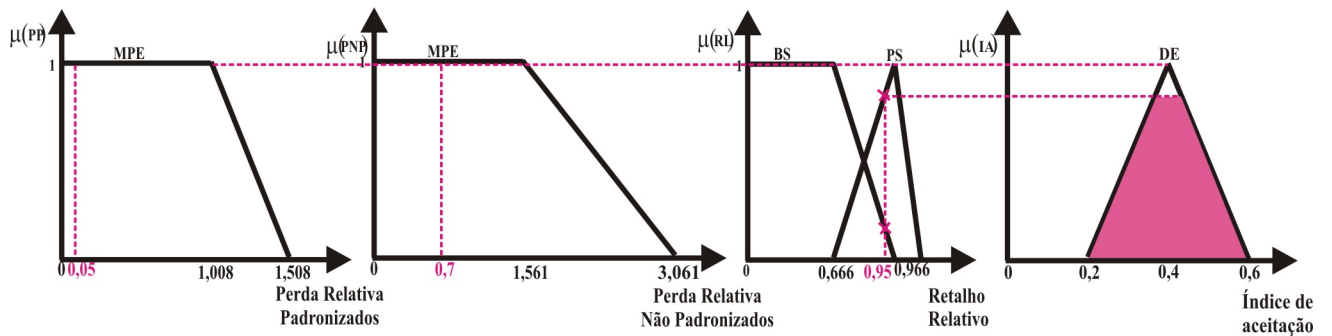


Figura 7.6: Região ativada: Regra 2.

Combinando as duas regiões *fuzzy* de saída, obtemos uma região *fuzzy* que representa a agregação das regras ativadas. Para esta agregação, na qual utilizamos a regra do máximo, temos a região apresentada na Figura 7.7.

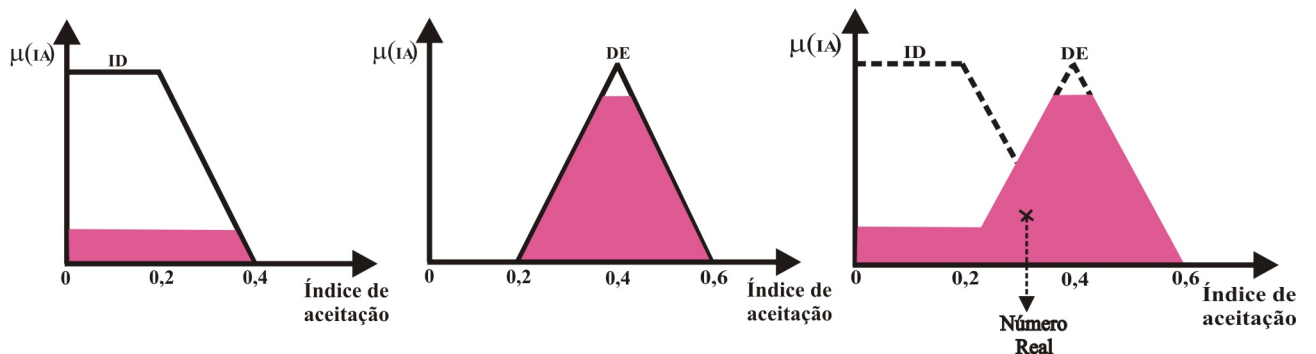


Figura 7.7: Ilustração gráfica do processo de Inferência *Fuzzy*: Saída.

Após obter a região resultante da agregação das regras ativadas, aplicamos um método de defuzzificação (CDA), o qual especifica um ponto na saída que melhor representa o conjunto *fuzzy*. O ponto localizado é um termo comum que combina a contribuição de cada regra ativada. A solução obtida pelo processo de defuzzificação é um número real que nos permite uma análise precisa do resultado.

Na Tabela 7.4, apresentamos o resultado e a classificação das heurísticas analisadas.

Tabela 7.4: Classificação das Soluções

| | | Construtiva | | Residual | |
|----------------|-------------|-------------|------------------------|--------------|--------------------------|
| | <i>COLA</i> | <i>FFD</i> | <i>FFD_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> |
| CDA | 0,8 | 0,437 | 0,388 | 0,6 | 0,385 |
| Solução | IND | AC | DE | AC | DE |

A partir das análises realizadas com o auxílio dos operadores *fuzzy*, temos que as melhores soluções para o problema de corte de estoque com sobras aproveitáveis quando relacionamos a perda relativa nos objetos padronizados e não padronizados e o retalho líquido relativo, foram apresentadas pelas heurísticas FFD_A e RAG_A . Observe que, embora não seja *ideal*, há uma melhora na solução quando comparamos as heurísticas clássicas com sua respectiva versão para o aproveitamento de sobras. Com relação ao algoritmo COLA, temos que ele apresenta uma solução *indesejável*. Isto ocorre pelo fato deste algoritmo apresentar uma perda relativa grande tanto para os objetos padronizados quanto para os objetos não padronizados.

As soluções apresentadas na Tabela 7.4 eram esperadas, pois os algoritmos utilizados foram os mesmos apresentados na Seção 4, o único diferencial foi o fato de considerarmos a baixa demanda dos itens. Com os testes realizados, foi possível obter de maneira mais simplificada a classificação dos procedimentos heurísticos, visto que existem características conflitantes em termos dos fatores que são relevantes para análise das soluções.

Esta técnica da lógica *fuzzy* foi aplicada nas soluções obtidas por alguns procedimentos heurísticos que não priorizam o corte dos objetos não padronizados (Seção 4), entretanto, outros procedimentos poderiam analisados, incluindo aqueles desenvolvidos para resolverem o problema de corte bidimensional com sobras aproveitáveis.

Capítulo 8

Conclusões e Propostas Futuras

Neste trabalho consideramos o problema de corte de estoque com sobras de material aproveitáveis, ou seja, toda sobra de material resultante de um objeto cortado, se grande o suficiente, pode ser utilizada para cortar futuras demandas de itens.

O estudo foi orientado para problemas de corte unidimensionais e bidimensionais, sendo que, para ambos os problemas, as abordagens são orientadas para a minimização das perdas. Desta forma, métodos e procedimentos heurísticos da literatura que têm como objetivo a minimização das perdas foram alterados para incluírem a possibilidade de geração de retalhos (sobras grandes), os quais não são contabilizados como perdas.

Para os problemas de corte unidimensionais, duas abordagens foram apresentadas: a primeira tem como principal objetivo a minimização das perdas geradas. Na segunda abordagem, também estendida para problemas bidimensionais, concomitante à minimização das perdas, os retalhos disponíveis em estoque têm prioridade de uso em relação aos demais objetos.

Os desempenhos dos procedimentos heurísticos desenvolvidos foram analisados considerando exemplares da literatura, problemas práticos e exemplares gerados aleatoriamente. Para os procedimentos heurísticos que priorizam o uso dos retalhos, também desenvolvemos uma simulação na qual sucessivos períodos são resolvidos, considerando os retalhos gerados em períodos anteriores e mantidos em estoque.

Com relação aos resultados obtidos para o problema de corte de estoque com sobras aproveitáveis, temos que, quando a minimização das perdas é o único objetivo,

as heurísticas RAG_A - versões 1, 2 e 3 apresentam soluções superiores em relação às heurísticas clássicas da literatura e as demais heurísticas de aproveitamento. Além disso, a heurística RAG_A - versão 2 apresentou soluções similares (em qualidade) às soluções exatas obtidas pelo modelo matemático (orientado ao item) de Abuabara e Morabito (2008). A solução via modelos matemáticos, como os propostos Abuabara e Morabito (2008) e (2009), é viável para problemas em que as quantidades de objetos e itens são moderadas, enquanto que as abordagens propostas neste trabalho permitem resolver problemas com grandes quantidades de objetos e itens, com alta repetição.

Ainda para os problemas de corte unidimensionais, quando impomos a condição de uso dos retalhos, observamos que a heurística RAG_A^P apresenta resultados melhores considerando os critérios estabelecidos que definem uma boa solução. Para os problemas de corte bidimensionais, a heurística RAG_A^{2D} apresentou um desempenho melhor quando a perda e a quantidade de retalhos são analisados simultaneamente. Em todos os casos analisados, observamos que as heurísticas desenvolvidas a partir da heurística RAG (Poldi e Arenales, 2009) geram as melhores soluções para o problema de corte de estoque com sobras aproveitáveis.

Devido à dificuldade em avaliar o desempenho dos procedimentos heurísticos que resolvem o problema de corte de estoque com sobras aproveitáveis, apresentamos uma técnica da lógica *fuzzy* que nos permitiu realizar a análise das soluções de maneira mais simplificada. Com os testes computacionais realizados, verificamos que os procedimentos heurísticos orientados para o problema de aproveitamento de sobras apresentam soluções superiores aos procedimentos heurísticos clássicos.

Como perspectiva e continuidade deste trabalho, merecem investigação os modelos matemáticos que consideram o problema de corte de estoque com sobras de material aproveitáveis no caso unidimensional. Para modelos orientados aos itens, os trabalhos de Abuabara e Morabito (2008) e (2009) podem ser abordados por técnicas bi-objetivo (perda \times retalhos). Para modelos orientados ao padrão, o trabalho de Scheithauer (1991), que define *a priori* os possíveis tamanhos para os retalhos, pode ser estendido. Este modelo também pode ser implementado utilizando outras técnicas de arredondamento para obtenção de uma solução inteira. Além disso, o modelo proposto também pode ser

reescrito como um modelo multiperíodos, no qual, os retalhos gerados seriam transferidos de um período para o outro. Uma outra abordagem seria definir um limitante inferior para o comprimento do retalho e a partir deste limitante, retalhos poderiam ser gerados com tamanho não estabelecidos *a priori*.

Para o problema de corte bidimensional com sobras aproveitáveis, uma possível alteração no grafo E/OU seria determinar o limitante inferior considerando padrões de corte em 2-estágios. Outras heurísticas residuais da literatura para a obtenção de uma solução inteira também podem ser investigadas e implementadas. Além disso a possibilidade de gerar itens para o estoque no lugar de determinados retalhos, também pode ser conveniente. Essa decisão cabe ao gerente de produção e a solução do problema de corte bidimensional com sobras aproveitáveis pode ser uma ferramenta de apoio à tomada de decisões.

Bibliografia

- [1] ABUABARA, A. , MORABITO R., *Modelos de programação inteira mista para o planejamento do corte unidimensional de tubos metálicos na indústria aeronáutica agrícola*. Gestão e Produção, 15: 605-617, (2008).
- [2] ABUABARA, A. , MORABITO R., *Cutting optimization of structural tubes to build agricultural light aircrafts*. Annals of Operations Research, 149: 149-165, (2009).
- [3] ALVAREZ-VALDÉS, R., PARAJÓN,A., TAMARIT.J., *A tabu search algorithm for large-scale guillotine (un)constrained two-dimensional cutting problems*. Computers and Operations Research, 29: 925-947, (2002).
- [4] ARBIB C., MARINELLI, F.,ROSSI, F., Di Iorio, F., *Cutting and reuse: The application from automobile component manufacturing*, Operations Research, 50: 923-934, (2002).
- [5] ARENALES, M. N., MORABITO, R., *An AND/OR - graph approach to the solution of two-dimensional non-guillotine cutting problems*. European Journal of Operational Research, 84: 599-617, (1995).
- [6] ARENALES, M. N., MORABITO, R., YANASSE H.(editores), *Cutting and packing problems*. Pesquisa Operacional, 19:107-299, (1999).
- [7] ARENALES, M. N., MORABITO, R., YANASSE, H. H., (editores), *O problema de corte e empacotamento*, Livro-texto de Mini curso. In: Congresso Nacional de Matemática Aplicada e Computacional, 26, 2004, São João del Rei - MG.

- [8] BEASLEY, J., *Algorithms for unconstrained two-dimensional gillotine cutting*. Journal of the Operational Research Society, 36: 297-306, (1985).
- [9] BISCHOFF, E., WÄSCHER, G. (editores), *Cutting and packing*. European Journal of Operational Research, v.84, n.3, special issue, (1995).
- [10] BROWN, A. R., *Optimum packing and depletion: the computer in space and resource usage proble*. New York: Macdonald - London and American Elsevier Inc, 1971.107p, (1971).
- [11] BURKE, E., KENDALL, G., WHITWELL, G., *A new placement heuristic for the orthogonal stock-cutting problem*. Operations Research, 52, 655-671, (2004).
- [12] CHERRI, A. C., ARENALES, . N., YANASSE, H. H., *The one dimensional cutting stock problems with usable leftover: A heuristic approach*. European Journal of Operational Research, 196: 897-908, (2009).
- [13] CHRISTOFIDES, N., WHITLOCK, C., *An algorithm for two-dimensional cutting problem*. Operations Research, 25: 30-44, (1977).
- [14] CHU, C., ANTONIO, J., *Approximation algorithms to solve real-life multicriteria cutting stock problems*. Operations Research, 47: 495-508, (1999).
- [15] CUI, Y. *Heuristic and exact algorithms for generating homogeneous constrained threestaged cutting patterns*. Computers and Operations Research, 35, 212-225, (2008).
- [16] CUNG, V., HIFI. M., LE CUN, B., *Constrained two-dimensional guillotine cutting stock problems: A best-first branch-and-bound algorithm*. International Transactions in Operational Research 7, 185-201, (2000).
- [17] DOWSLAND, K., DOWSLAND, W., *Packing problems*. European Journal of Operational Research, 56: 2-14, (1992).
- [18] DOWSLAND, K., HERBERT, E.; KENDALL, G., BURKE, E., *Using tree search bounds to enhance a genetic algorithm approach to two rectangle packing problems*. European Journal of Operational Research, 168, 390-402, (2006).

- [19] DYCKHOFF, H., *A new linear programming approach to the cutting stock problem*. Operations Research, 29: 1092-1104, (1981).
- [20] DYCKHOFF, H., KRUSE, H. J., ABEL, D., GAL, T., *Trim loss and related problems*. The International Journal of Management Science, 13: 59-72, (1985).
- [21] DYCKHOFF, H., WÄSCHER, G. (editores), *Cutting and packing*. European Journal of Operational Research, v.44, n.2, special issue, (1990).
- [22] DYCKHOFF, H., *A typology of cutting and packing problems*. European Journal Operational Research, 44: 145-159, (1990).
- [23] DYCKHOFF H., FINKE U. *Cutting and packing in production and distribution: typology and bibliography*. Heidelberg: Springer, (1992).
- [24] DYCKHOFF, H., SCHEITHAUER, G., TERNO, J., *Cutting and packing*. In: Mico, M., Maffioli, F., Martello, S. (Ed.) *An noted bibliographies in ombinatorial optimization*. New York: John Wiley & Sons. p.393-414, (1997).
- [25] ESICUP - Euro Special Interest Group on Cutting and Packing. Available in: <<http://www.apdio.pt/esicup/>> (accessed in 2008).
- [26] GILMORE, P. C., GOMORY, R. E., *A linear programming approach to the cutting stock problem*. Operations Research, 9: 848-859, (1961).
- [27] GILMORE, P. C., GOMORY, R. E., *A linear programming approach to the cutting stock problem - Part II*. Operations Research, 11: 863-888, (1963).
- [28] GILMORE, P. C., GOMORY, R. E., *Multi-stage cutting stock problems of two and more dimensions*. Operations Research, 13: 94-120, (1965).
- [29] GRADISAR, M., JESENKO, J., RESINOVIC, C., *Optimization of roll cutting in clothing industry*. Computers & Operational Research, 10: 945-953, (1997).
- [30] GRADISAR, M., KLJAJIC, M., RESINOVIC, C., JESENKO, J., *A sequential heuristic procedure for one-dimentional cutting*. European Journal of Operational Research, 114: 557-568, (1999a).

- [31] GRADISAR, M., RESINOVIC, C., KLJAJIC, M., *A hybrid approach for optimization of one-dimensional cutting*. European Journal of Operational Research, 119: 719-728, (1999b).
- [32] GRADISAR, M., TRKMAN, P., *A combined approach to the solution to the general one-dimensional cutting stock problem*. Computers & Operations Research, 32: 1793-1807, (2005).
- [33] HAESSLER, R. W., *Controlling cutting pattern changes in one-dimensional trim loss problems*. Operations Research, 23: 483-493, (1975).
- [34] HAESSLER, R. W., *A note on computational modifications to the Gilmore-Gomory cutting stock algorithm*. Operations Research, 28: 1001-1005, (1980).
- [35] HERZ, J., *Recursive computational procedure for two-dimensional stock cutting*. IBM Journal of Research and Development, 16: 462-469, (1972).
- [36] HINXMAN, A., *The trim-loss and assortment problems: a survey*. European Journal of Operational Research, 5: 8-18, (1980).
- [37] HIFI, M., *An improvement of Viswanathan and Bagchi's exact algorithm for constrained two-dimensional cutting stock*. Computers and Operations Research 24, 727-736, (1997).
- [38] HIFI, M. (editores), *Special issue: Cutting and packing problems*. Studia Informatica Universalis, 2, 1, 1-161, (2002).
- [39] HIFI, M., *Dynamic programming and hill-climbing techniques for constrained two-dimensional cutting stock problems*. Journal of Combinatorial Optimization 8, 65-84, (2004).
- [40] KANTOROVICH, L.V., *Mathematical methods of organizing and planning production*. Management Science, 6: 366-422, (1960).
- [41] KOCH, S., KÖNIG, S., WÄSCHER, G., *Linear programming for a cutting problem in the wood processing industry - a case study*. Working Paper n° 14, FEMM, (2008).

- [42] KOS, L., DUHOVNIK, J., *Cutting optimization with variable-sized stock and inventory status data*. International Journal of Production Research, 40: 2289-2301, (2002).
- [43] LODI, A., MARTELLO, S., MONACI, M., *Two-dimensional packing problems: a survey*. European Journal of Operational Research, 141, 241-252, (2002).
- [44] LUENBERGER, D.G., *Linear and Nonlinear Programming*. 2.ed. Reading Mass.: Addison-Wesley, 1989, c1984.491p.
- [45] MACEDO, R., SILVA, E., ALVES, C., ALVELOS, F., CARVALHO, J. V., ARBIB, C., MARINELLI, F., PEZZELLA, F., DI GIOVANNI, L., GAMBELLA, L., *2D Cutting Stock Optimization Software Survey*, submitted to OR/MS Today, (2008).
- [46] MARTELLO, S., TOTH, P., Knapsack Problems. Chichester: Wiley, 1990.296p.
- [47] MARTELLO, S. (ed.) , Special issue: Knapsack, packing and cutting, Part I: One dimensional knapsack problems. INFOR, 32, 3, (1994a).
- [48] MARTELLO, S.,(ed.) , Special issue: Knapsack, packing and cutting, Part II: Multidimensional knapsack and cutting stock problems. INFOR, 32, 4, (1994b).
- [49] MORABITO, R., *Corte de estoque bidimensional*. Dissertação de Mestrado, ICMC - USP, SP, Brasil (1989).
- [50] MORABITO, R., *Uma abordagem em grafo E/OU para o problema de empacotamento: Aplicação ao carregamento de paletes e contêineres*. Tese de Doutorado, EESC - USP, (1992).
- [51] MORABITO, R., Problemas de corte e empacotamento. Livro-texto de Mini curso, Elavio, Montevidéo - Uruguai, (2004).
- [52] MORABITO, R., ARENALES, M. N., *An AND/OR - graph approach to the container loading problem*. International Transactions in Operational Research, 1: 59-73, (1994).

- [53] MORABITO, R., ARENALES, M. N., *Staged and constrained two-dimensional guillotine cutting problems: An AND/OR - graph approach*. European Journal of Operational Research, 94: 548-560, (1996).
- [54] MORABITO, R., ARENALES, M. N., ARCARO, V. F., *AND-OR-graph approach for two-dimensional cutting problems*. European Journal of Operational Research, 58: 263-271, (1992).
- [55] MORABITO R., PUREZA V., *A heuristic approach based on dynamic programming and and/or-graph search for the constrained two-dimensional guillotine cutting problem*. Annals of Operations Research (2008), doi: 10.1007/s10479-008-0457-4.
- [56] OLIVEIRA, J., FERREIRA, J., *An improved version of Wang's algorithm for two-dimensional cutting problems*. European Journal of Operational Research, 44: 256-266, (1990).
- [57] OLIVEIRA, J. F., WÄSCHER, G. (editores) *Special Issue on Cutting and Packing*. European Journal of Operational Research, 183, (2007).
- [58] PARADA, V., ALVARENGA, A. G., DIEGO, J., *Exact solutions for constrained two-dimensional cutting problems*. European Journal of Operational Research 84: 633-644 (1995).
- [59] PEDRYCZ, W., GOMIDE, F. C., *An Introduction to Fuzzy Sets: Analysis and Design*. The MIT Press, Massachusetts, 1998.
- [60] POLDI, K. C., *O problema de corte de estoque multiperíodo*. Tese de Doutorado, ICMC - USP, SP, Brasil (2007).
- [61] POLDI, K. C., ARENALES, M. N., *Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths*. Computers and Operations Research, 36: 2074-2081, (2009).
- [62] POLDI, K. C., ARENALES, M. N., VIANNA, A. C. G. *Two-dimensional cutting stock problem under low demand: a case study*. Operations Research Proceedings

- 2005 - Selected papers of the German Operations Research Society (GOR) 291-296.
- [63] RIEHME J., SCHEITHAUER G., TERNO J. *The solution of two-stage guillotine cuttingstock problems having extremelly varying order demands*. European Journal of Operational Research, 91: 543-552, (1996).
- [64] ROODMAN, G. M., *Near-optimal solutions to one-dimensional cutting stock problem*. Computers and Operations Research, 13: 713-719, (1986).
- [65] SCHEITHAUER, G., *A note on handling residual length*. Optimization, 22: 461-466, (1991).
- [66] SINUANY-STERM, Z., WEINER I., *The one dimensional cutting stock problem using two objectives*. Journal of Operations Research Society, 45: 231-236, (1994).
- [67] STADTLER, H., *A one-dimensional cutting stock problem in the Aluminium Industry and its solution*. European Journal of Operational Research, 44: 209-223, (1990).
- [68] SWEENEY, P., PARTERNOSTER E., *Cutting and packing problems: a categorized, application-oriented research bibliography*. Journal of Operational Research Society, 43: 691-706, (1992).
- [69] TRKMAN, P., Comunicação Privada (09/11/2005).
- [70] TRKMAN, P., GRADISAR, M., *One-dimensional cutting stock optimization in consecutive time periods*. European journal of Operational Research, 179: 291-301, (2007)
- [71] VAHRENKAMP, R., *Random search in the one-dimensional cutting stock problem*. European Journal of Operational Research, 95: 191-200, (1996).
- [72] VIANNA, A. C. G., *Problemas de corte e empacotamento: Uma abordagem em grafo E/OU*. Tese de Doutorado, ICMC - USP, SP, Brasil (2000).

- [73] VISWANATHAN, K. V.; BAGCHI, A., Best-first search methods for constrained two-dimensional cutting stock problems. *Operations Research* 41(4), 768-776, (1993).
- [74] WANG, D. X, A course in Fuzzy Systems and control. Prentice-Hall, Inc., Upper Saddle River, New Jersey, (1996).
- [75] WANG, P., *Two algorithms for constrained two-dimensional cutting stock problems. Operations Research*, 31: 573-587, (1983).
- [76] WANG, P.Y., WÄSCHER, G., *Cutting and packing. European Journal of Operational Research*, 141: 239-469, (2002).
- [77] WÄSCHER, G., GAU, T., *Heuristics for the integer one-dimensional cutting stock problem: a computational study. OR Spektrum*, 18: 131-144, (1996).
- [78] WÄSCHER, G., HAUßNER, H., SCHUMANN, H., *An improved typology cutting and packing problems". European Journal of Operational Research*, 183:1109-1130, (2007).

Apêndice A - Resultados computacionais

Neste apêndice, apresentamos todas as tabelas com os testes computacionais realizados na Seção 6.1.2 e 7.3.

Experimentos computacionais - Seção 6.1.2

Nesta seção do Apêndice A apresentamos o resultado médio obtido por cada classe de exemplos geradas na Seção 6.1.2.

Tabela A.1: Número médio de objetos não padronizados utilizados

| | | Construtiva | | | | Residual | | | | | | | | | |
|-----------------|-------------|-------------|------------------------|---------------|---------------------------|------------|------------------------|---------------|---------------------------|--------------|--------------------------|--------------|--------------------------|--------------|--------------------------|
| | <i>COLA</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> | <i>RAG 2</i> | <i>RAG_A 2</i> | <i>RAG 3</i> | <i>RAG_A 3</i> |
| C ₁ | 17.7 | 11.0 | 8.0 | 7.5 | 8.6 | 8.9 | 7.7 | 5.4 | 5.8 | 2.8 | 4.8 | 3.9 | 4.9 | 3.4 | 4.5 |
| C ₂ | 16.0 | 8.5 | 4.5 | 4.3 | 4.5 | 4.9 | 4.5 | 4.7 | 5.1 | 3.8 | 4.2 | 4.2 | 4.4 | 3.2 | 4.3 |
| C ₃ | 18.1 | 9.2 | 8.8 | 7.8 | 8.7 | 9.5 | 6.0 | 4.8 | 4.6 | 3.9 | 4.5 | 4.3 | 4.5 | 3.8 | 4.5 |
| C ₄ | 18.0 | 8.7 | 8.0 | 8.1 | 4.4 | 7.1 | 4.7 | 4.0 | 2.9 | 3.7 | 4.1 | 3.9 | 4.2 | 2.4 | 3.5 |
| C ₅ | 18.1 | 8.8 | 6.7 | 6.7 | 5.5 | 7.8 | 5.6 | 5.4 | 4.6 | 3.2 | 3.9 | 3.5 | 3.9 | 3.9 | 4.2 |
| C ₆ | 15.5 | 10.0 | 7.0 | 8.9 | 6.1 | 6.4 | 4.3 | 3.7 | 3.6 | 2.3 | 3.4 | 2.8 | 3.4 | 2.8 | 3.0 |
| C ₇ | 30.4 | 10.8 | 7.0 | 8.1 | 11.7 | 9.5 | 5.9 | 4.8 | 5.1 | 4.8 | 4.6 | 5.2 | 4.6 | 3.8 | 4.8 |
| C ₈ | 26.5 | 11.2 | 10.6 | 8.8 | 11.2 | 9.2 | 6.0 | 4.8 | 4.8 | 3.8 | 6.2 | 4.1 | 6.1 | 3.5 | 5.4 |
| C ₉ | 30.1 | 11.3 | 7.1 | 7.9 | 9.9 | 9.5 | 7.6 | 5.0 | 5.1 | 4.2 | 4.2 | 4.5 | 4.2 | 3.5 | 4.3 |
| C ₁₀ | 30.0 | 11.6 | 10.3 | 6.2 | 11.6 | 11.1 | 5.7 | 2.6 | 4.1 | 3.8 | 4.1 | 3.8 | 4.2 | 2.7 | 3.5 |
| C ₁₁ | 30.1 | 7.4 | 6.9 | 7.0 | 8.7 | 6.8 | 7.1 | 5.4 | 6.4 | 3.1 | 3.8 | 3.3 | 3.7 | 4.7 | 6.2 |
| C ₁₂ | 28.9 | 14.2 | 9.9 | 7.8 | 10.1 | 10.7 | 3.7 | 4.6 | 4.9 | 2.3 | 2.6 | 2.2 | 2.6 | 3.1 | 3.8 |
| C ₁₃ | 42.8 | 11.1 | 8.0 | 9.0 | 12.0 | 9.7 | 5.7 | 5.4 | 5.5 | 4.2 | 4.4 | 4.6 | 4.4 | 4.2 | 3.6 |
| C ₁₄ | 39.6 | 14.6 | 12.5 | 10.5 | 13.5 | 10.4 | 5.4 | 4.2 | 4.8 | 4.8 | 5.6 | 4.8 | 6.0 | 4.2 | 6.8 |
| C ₁₅ | 43.9 | 11.6 | 9.4 | 11.1 | 13.5 | 9.7 | 6.7 | 4.9 | 4.9 | 3.9 | 4.0 | 4.3 | 4.2 | 4.2 | 5.0 |
| C ₁₆ | 43.7 | 15.2 | 12.6 | 14.6 | 14.3 | 13.9 | 6.4 | 9.7 | 5.0 | 4.1 | 4.9 | 4.1 | 4.9 | 3.1 | 3.8 |
| Média | 28.1 | 11.0 | 8.6 | 8.4 | 9.6 | 9.1 | 5.8 | 5.0 | 4.8 | 3.7 | 4.3 | 4.0 | 4.4 | 3.5 | 4.5 |

Tabela A.2: Comprimento médio de objetos não padronizados utilizados

| | | Construtiva | | | | Residual | | | | | | | | | |
|-----------------|----------------|-------------|------------------------|---------------|---------------------------|------------|------------------------|---------------|---------------------------|---------------|--------------------------|--------------|--------------------------|---------------|--------------------------|
| | <i>COLA</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> | <i>RAG 2</i> | <i>RAG_A 2</i> | <i>RAG 3</i> | <i>RAG_A 3</i> |
| C ₁ | 3000.9 | 1519.4 | 983.3 | 1170.3 | 1475.9 | 1184.2 | 873.3 | <i>586.4</i> | 733.6 | 589.0 | 796.5 | 661.1 | 840.9 | 758.2 | 900.8 |
| C ₂ | 5050.7 | 2853.0 | 1662.5 | 1492.6 | 1765.7 | 1626.3 | 1745.6 | 1609.2 | 2006.1 | 1342.5 | 1581.5 | 1486.1 | 1622.2 | <i>1113.4</i> | 1595.5 |
| C ₃ | 3447.0 | 1496.3 | 1165.4 | 1054.0 | 1620.5 | 1256.3 | <i>604.9</i> | 593.1 | 669.7 | 653.3 | 718.1 | 686.5 | 761.9 | 701.1 | 796.5 |
| C ₄ | 5812.7 | 2934.7 | 2778.5 | 2858.5 | 1740.2 | 2228.8 | 1715.6 | 1403.1 | 1076.7 | 1224.5 | 1448.4 | 1318.7 | 1515.3 | <i>836.7</i> | 1276.6 |
| C ₅ | 3474.3 | 1175.3 | 702.9 | 969.0 | 1225.4 | 1004.3 | 681.6 | 659.9 | 746.0 | <i>542.3</i> | 646.3 | 544.8 | 649.1 | 593.2 | 621.2 |
| C ₆ | 5509.5 | 3456.6 | 2446.3 | 2446.3 | 2630.1 | 2088.1 | 1392.1 | 1247.9 | 1504.5 | <i>735.5</i> | 1152.1 | 897.8 | 1152.1 | 902.7 | 1093.9 |
| C ₇ | 7585.0 | 1998.5 | 644.5 | 1393.5 | 2313.1 | 1412.0 | 607.3 | 496.7 | 547.7 | <i>678.4</i> | 683.7 | 681.4 | 678.6 | 708.8 | 837.4 |
| C ₈ | 7653.4 | 3042.3 | 3074.8 | 2443.8 | 3877.8 | 2333.4 | 1703.8 | 1287.7 | 1488.1 | <i>1071.3</i> | 1874.2 | 1175.8 | 1810.5 | 1117.8 | 1644.5 |
| C ₉ | 6414.6 | 1655.6 | 629.1 | 1076.7 | 1651.2 | 1134.9 | 761.7 | <i>540.1</i> | 578.5 | 732.2 | 781.5 | 742.6 | 783.2 | 628.2 | 668.1 |
| C ₁₀ | 8637.5 | 3472.7 | 2899.3 | 1997.2 | 4362.2 | 2861.2 | 1606.2 | 797.5 | 1417.4 | 1116.3 | 1319.4 | 1105.7 | 1356.7 | <i>770.3</i> | 1139.1 |
| C ₁₁ | 4975.3 | 812.4 | 498.6 | 645.3 | 1369.1 | 670.5 | 488.7 | <i>417.7</i> | 506.4 | 449.5 | 511.2 | 485.4 | 494.6 | 727.6 | 871.1 |
| C ₁₂ | 9454.3 | 4533.2 | 2997.3 | 2761.2 | 3948.4 | 3365.3 | 1033.3 | 1372.9 | 1725.2 | 682.9 | 776.6 | <i>681.4</i> | 786.0 | 1037.3 | 1217.1 |
| C ₁₃ | 10433.4 | 1489.0 | 839.4 | 1323.6 | 1828.3 | 1312.0 | 579.3 | <i>415.4</i> | 515.3 | 673.0 | 674.2 | 685.5 | 675.6 | 709.5 | 738.8 |
| C ₁₄ | 11072.6 | 3347.3 | 3824.1 | 2825.9 | 4115.0 | 2331.4 | 1249.9 | <i>968.8</i> | 1120.0 | 1266.6 | 1522.7 | 1270.6 | 1615.0 | 1103.4 | 1678.7 |
| C ₁₅ | 9559.6 | 1705.8 | 1058.4 | 1512.9 | 2057.0 | 1408.6 | 618.5 | <i>439.9</i> | 461.7 | 646.8 | 689.6 | 658.8 | 694.0 | 586.1 | 738.1 |
| C ₁₆ | 10964.1 | 3627.1 | 2921.5 | 3876.4 | 4185.0 | 3015.1 | 1267.9 | 1754.6 | 1137.6 | 1056.8 | 1230.4 | 1075.2 | 1230.4 | <i>518.2</i> | 1023.0 |
| Média | 7065.3 | 2445.0 | 1820.4 | 1865.5 | 2510.3 | 1827.0 | 1058.1 | 911.9 | 1014.7 | 841.3 | 1025.4 | 884.8 | 1041.6 | <i>800.9</i> | 1052.5 |

Tabela A.3: Número médio de objetos padronizados utilizados

| | COLA | Construtiva | | | | Residual | | | | | | | | | |
|-----------------|--------------|-------------|------------------|--------------|---------------------|--------------|------------------|--------------|---------------------|--------------|--------------------|--------------|--------------------|--------------|--------------------|
| | | FFD | FFD _A | Gulosa | Gulosa _A | FFD | FFD _A | Gulosa | Gulosa _A | RAG 1 | RAG _A 1 | RAG 2 | RAG _A 2 | RAG 3 | RAG _A 3 |
| C ₁ | 108.5 | 107.7 | 108.3 | 110.3 | 110.3 | <i>102.5</i> | 102.9 | 103.0 | 103.0 | 102.6 | 102.6 | <i>102.5</i> | 102.6 | 102.6 | 102.7 |
| C ₂ | 114.2 | 111.0 | 123.3 | 116.7 | 129.5 | <i>110.3</i> | 110.6 | 110.4 | 110.7 | 110.4 | 113.7 | <i>110.3</i> | 113.3 | 110.5 | 113.6 |
| C ₃ | 107.3 | 105.5 | 108.0 | 109.6 | 109.4 | 101.5 | 102.1 | 102.1 | 102.1 | <i>101.3</i> | 101.5 | <i>101.3</i> | 101.5 | <i>101.3</i> | 101.5 |
| C ₄ | 102.5 | 100.5 | 103.9 | 105.8 | 116.4 | <i>99.1</i> | 100.1 | 100.1 | 101.0 | 100.0 | 101.4 | 99.9 | 101.1 | 100.3 | 101.5 |
| C ₅ | 99.35 | 103.9 | 106.0 | 106.3 | 106.2 | 98.5 | 98.8 | 98.8 | 98.8 | 98.0 | 98.0 | 98.0 | 98.0 | <i>97.9</i> | 98.0 |
| C ₆ | 106.7 | 105.6 | 109.0 | 108.7 | 113.2 | <i>102.0</i> | 103.0 | 102.8 | 103.0 | 102.6 | 102.5 | 102.5 | 102.5 | 102.6 | 102.8 |
| C ₇ | 103.1 | 105.6 | 107.9 | 108.6 | 108.2 | <i>101.9</i> | 102.8 | 102.9 | 102.9 | 102.2 | 102.4 | 102.2 | 102.4 | 102.3 | 102.3 |
| C ₈ | 107.3 | 109.0 | 111.4 | 113.1 | 114.4 | <i>106.7</i> | 107.5 | 107.9 | 107.8 | 107.7 | 107.5 | 107.6 | 107.5 | 107.8 | 107.6 |
| C ₉ | 104.9 | 107.7 | 109.8 | 109.6 | 109.4 | 101.9 | 102.3 | 102.6 | 102.7 | <i>101.8</i> | <i>101.8</i> | <i>101.8</i> | <i>101.8</i> | <i>101.8</i> | 101.9 |
| C ₁₀ | 98.1 | 100.0 | 102.0 | 105.2 | 104.2 | <i>96.3</i> | 97.7 | 98.3 | 98.0 | 97.6 | 97.6 | 97.6 | 97.6 | 98.0 | 97.9 |
| C ₁₁ | 105.1 | 107.5 | 108.7 | 108.6 | 108.3 | <i>101.3</i> | 101.6 | 101.6 | 101.8 | 100.7 | 100.8 | 100.7 | 100.8 | 100.4 | 100.4 |
| C ₁₂ | 101.1 | 102.0 | 105.5 | 107.2 | 107.2 | <i>98.6</i> | 101.0 | 100.8 | 100.6 | 100.5 | 100.5 | 100.5 | 100.5 | 100.3 | 100.3 |
| C ₁₃ | 104.2 | 109.8 | 111.8 | 112.4 | 112.5 | <i>105.7</i> | 106.3 | 106.4 | 106.6 | 105.8 | 106.0 | 105.8 | 106.0 | <i>105.7</i> | 105.8 |
| C ₁₄ | <i>92.3</i> | 99.0 | 98.1 | 99.1 | 101.0 | 93.4 | 95.1 | 95.3 | 95.3 | 94.7 | 94.6 | 94.7 | 94.6 | 94.9 | 94.7 |
| C ₁₅ | <i>104.7</i> | 110.1 | 111.9 | 112.1 | 111.9 | 104.9 | 105.8 | 106.0 | 106.0 | 105.0 | 105.1 | 105.0 | 105.1 | 105.0 | 105.1 |
| C ₁₆ | 103.9 | 108.1 | 110.3 | 110.6 | 112.7 | 103.7 | 105.5 | <i>85.9</i> | 105.7 | 105.2 | 105.3 | 105.2 | 105.3 | 105.5 | 105.4 |
| Média | 104.0 | 105.8 | 108.5 | 109.0 | 110.9 | 101.8 | 102.7 | <i>101.6</i> | 102.9 | 102.3 | 102.6 | 102.2 | 102.5 | 102.3 | 102.6 |

Tabela A.4: Comprimento médio de objetos padronizados utilizados

| | <i>COLA</i> | Construtiva | | | | Residual | | | | | | | | | |
|-----------------------|-----------------|--------------------|------------------------|---------------|---------------------------|-----------------|------------------------|---------------|---------------------------|---------------|--------------------------|---------------|--------------------------|--------------|--------------------------|
| | | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> | <i>RAG 2</i> | <i>RAG_A 2</i> | <i>RAG 3</i> | <i>RAG_A 3</i> |
| <i>C₁</i> | <i>109545</i> | 111380 | 111525 | 111630 | 111750 | 111360 | 111695 | 111705 | 111705 | 111570 | 111620 | 111525 | 111625 | 111450 | 111530 |
| <i>C₂</i> | <i>116025</i> | 117590 | 127955 | 120065 | 135655 | 116680 | 116900 | 116665 | 116985 | 116725 | 120010 | 116605 | 119495 | 116945 | 119875 |
| <i>C₃</i> | <i>108170</i> | 110175 | 110460 | 110600 | 110495 | 110300 | 110855 | 110690 | 110740 | 110480 | 110680 | 110480 | 110630 | 110465 | 110605 |
| <i>C₄</i> | <i>103440</i> | 106350 | 107545 | 108010 | 120930 | 105940 | 106885 | 106770 | 107760 | 106870 | 108195 | 106760 | 107920 | 107285 | 108370 |
| <i>C₅</i> | <i>104765</i> | 107085 | 107620 | 107130 | 107090 | 107160 | 107385 | 107165 | 107215 | 107220 | 107270 | 107220 | 107270 | 107185 | 107285 |
| <i>C₆</i> | <i>107510</i> | 109870 | 110980 | 110295 | 115945 | 110830 | 111675 | 111435 | 111710 | 111910 | 111755 | 111750 | 111755 | 111775 | 112025 |
| <i>C₇</i> | <i>104190</i> | 109705 | 111170 | 110270 | 109910 | 110040 | 110975 | 110940 | 110940 | 110570 | 110770 | 110570 | 110770 | 110545 | 110590 |
| <i>C₈</i> | <i>108640</i> | 114310 | 115065 | 115170 | 116995 | 113205 | 113980 | 114375 | 114270 | 114235 | 113995 | 114140 | 114010 | 114235 | 114115 |
| <i>C₉</i> | <i>105515</i> | 110335 | 111430 | 110815 | 110540 | 110655 | 110965 | 111125 | 111275 | 110840 | 110890 | 110840 | 110890 | 110930 | 111030 |
| <i>C₁₀</i> | <i>98455</i> | 104440 | 104505 | 106895 | 107045 | 103885 | 105230 | 105735 | 105485 | 105375 | 105380 | 105380 | 105385 | 105735 | 105590 |
| <i>C₁₁</i> | <i>105885</i> | 110010 | 110285 | 109620 | 109290 | 109935 | 110155 | 109950 | 110150 | 109880 | 109980 | 109880 | 110030 | 109595 | 109645 |
| <i>C₁₂</i> | <i>101590</i> | 106600 | 107855 | 108440 | 108575 | 107370 | 109650 | 109285 | 109135 | 109815 | 109815 | 109815 | 109815 | 109520 | 109520 |
| <i>C₁₃</i> | <i>105155</i> | 114130 | 114750 | 114030 | 114195 | 114110 | 114755 | 114745 | 114945 | 114435 | 114635 | 114435 | 114635 | 114425 | 114475 |
| <i>C₁₄</i> | <i>92775</i> | 101790 | 100645 | 101270 | 103440 | 100935 | 102115 | 102315 | 102320 | 101860 | 101750 | 101865 | 101710 | 102045 | 101780 |
| <i>C₁₅</i> | <i>105655</i> | 113610 | 114230 | 113655 | 113505 | 113610 | 114535 | 114485 | 114485 | 114150 | 114200 | 114200 | 114200 | 114220 | 114320 |
| <i>C₁₆</i> | <i>104620</i> | 112270 | 112570 | 112500 | 114995 | 112165 | 113830 | 114150 | 113950 | 113865 | 113970 | 113970 | 113970 | 114125 | 114075 |
| Média | <i>105120.9</i> | 109978.1 | 111161.9 | 110649.7 | 113147.2 | 109886.3 | 110724.1 | 110720.9 | 110816.9 | 110612.5 | 110932.2 | 110589.7 | 110881.9 | 110655.0 | 110926.9 |

Tabela A.5: Perda média

| | COLA | Construtiva | | | | Residual | | | | | | | | | |
|-----------------|-------|---------------|------------------|---------------|---------------------|----------|------------------|--------|---------------------|-------|--------------------|-------|--------------------|-------|--------------------|
| | | FFD | FFD _A | Gulosa | Gulosa _A | FFD | FFD _A | Gulosa | Gulosa _A | RAG 1 | RAG _A 1 | RAG 2 | RAG _A 2 | RAG 3 | RAG _A 3 |
| C ₁ | 17.5 | 284.3 | 26.4 | 158.6 | 15.0 | 29.9 | 5.4 | 20.7 | 4.3 | 16.0 | 7.8 | 31.9 | 7.1 | 22.3 | 6.6 |
| C ₂ | 925.9 | 2406.4 | 98.7 | 2712.7 | 40.2 | 326.3 | 157.2 | 331.7 | 156.3 | 323.3 | 87.6 | 379.1 | 97.3 | 328.9 | 93.8 |
| C ₃ | 4.2 | 105.1 | 12.7 | 163.5 | 12.6 | 10.9 | 3.7 | 47.1 | 2.6 | 28.6 | 3.2 | 31.5 | 3.3 | 22.2 | 3.4 |
| C ₄ | 371.5 | 889.5 | 69.4 | 1850.9 | 32.3 | 168.7 | 35.6 | 217.4 | 27.9 | 138.8 | 18.8 | 123.0 | 18.7 | 165.9 | 24.4 |
| C ₅ | 0.8 | 44.5 | 6.2 | 132.4 | 8.5 | 7.1 | 1.7 | 64.0 | 4.2 | 25.9 | 3.5 | 32.5 | 3.5 | 26.0 | 1.7 |
| C ₆ | 24.0 | 354.9 | 28.4 | 531.0 | 74.5 | 47.9 | 5.7 | 133.4 | 11.9 | 71.6 | 7.4 | 75.8 | 7.4 | 114.7 | 6.7 |
| C ₇ | 6.4 | 118.8 | 31.6 | 55.1 | 11.0 | 15.2 | 1.9 | 8.9 | 1.4 | 18.4 | 2.4 | 19.1 | 2.4 | 12.3 | 3.3 |
| C ₈ | 297.0 | 1630.9 | 102.5 | 795.6 | 54.3 | 89.3 | 9.4 | 63.4 | 10.9 | 69.7 | 19.4 | 79.2 | 17.0 | 84.9 | 21.4 |
| C ₉ | 2.3 | 57.1 | 21.6 | 42.3 | 11.1 | 10.0 | 3.1 | 17.3 | 3.3 | 12.3 | 1.2 | 13.7 | 1.2 | 9.6 | 1.9 |
| C ₁₀ | 136.9 | 951.9 | 56.4 | 1124.2 | 20.1 | 120.0 | 7.1 | 58.6 | 4.4 | 96.7 | 8.1 | 98.7 | 7.4 | 84.0 | 8.8 |
| C ₁₁ | 0.2 | 13.0 | 9.1 | 44.7 | 11.7 | 3.6 | 1.5 | 20.5 | 2.5 | 9.4 | 1.6 | 10.1 | 1.7 | 22.3 | 3.9 |
| C ₁₂ | 13.5 | 169.5 | 11.9 | 294.1 | 37.8 | 26.4 | 2.3 | 96.3 | 6.8 | 39.5 | 4.0 | 36.3 | 3.8 | 45.2 | 2.8 |
| C ₁₃ | 5.8 | 98.0 | 17.1 | 76.0 | 16.3 | 8.2 | 1.8 | 15.7 | 3.1 | 8.3 | 1.1 | 8.4 | 1.0 | 10.4 | 1.1 |
| C ₁₄ | 155.0 | 1460.3 | 63.0 | 593.4 | 49.6 | 52.1 | 6.5 | 43.6 | 7.7 | 50.9 | 11.5 | 46.1 | 12.0 | 55.0 | 12.5 |
| C ₁₅ | 1.5 | 36.6 | 9.5 | 42.1 | 20.9 | 4.2 | 1.1 | 9.7 | 1.9 | 9.1 | 1.4 | 11.2 | 1.3 | 10.6 | 2.0 |
| C ₁₆ | 58.2 | 614.4 | 29.0 | 592.0 | 50.6 | 61.7 | 6.1 | 33.8 | 11.3 | 43.4 | 4.3 | 48.5 | 4.3 | 44.3 | 5.6 |
| Média | 126.3 | 577.2 | 37.1 | 575.5 | 29.2 | 61.3 | 15.6 | 73.9 | 16.3 | 60.1 | 11.5 | 65.3 | 11.8 | 66.1 | 12.5 |

Tabela A.6: Comprimento médio da perda nos objetos padronizados

| | COLA | Construtiva | | | | Residual | | | | | | | | | |
|-----------------|-------|---------------|------------------|---------------|---------------------|----------|------------------|--------|---------------------|-------|--------------------|-------|--------------------|-------|--------------------|
| | | FFD | FFD _A | Gulosa | Gulosa _A | FFD | FFD _A | Gulosa | Gulosa _A | RAG 1 | RAG _A 1 | RAG 2 | RAG _A 2 | RAG 3 | RAG _A 3 |
| C ₁ | 17.3 | 271.2 | 21.5 | 89.2 | 8.0 | 3.8 | 0.3 | 0.4 | 0.0 | 3.5 | 0.2 | 2.0 | 0.1 | 1.9 | 0.4 |
| C ₂ | 840.3 | 2339.1 | 93.1 | 2481.4 | 34.5 | 115.8 | 153.6 | 164.9 | 151.6 | 202.5 | 81.8 | 208.8 | 88.5 | 224.7 | 87.4 |
| C ₃ | 4.2 | 100.3 | 10.1 | 66.1 | 7.4 | 2.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.3 | 0.1 |
| C ₄ | 353.2 | 837.5 | 57.2 | 1397.8 | 24.6 | 53.7 | 25.7 | 31.4 | 24.3 | 31.8 | 11.5 | 26.9 | 11.6 | 74.0 | 15.7 |
| C ₅ | 0.8 | 43.9 | 5.4 | 10.5 | 2.9 | 3.5 | 0.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| C ₆ | 24.0 | 337.2 | 21.2 | 314.6 | 31.2 | 0.7 | 0.7 | 5.2 | 1.5 | 6.7 | 0.1 | 1.3 | 0.1 | 15.6 | 0.6 |
| C ₇ | 5.7 | 112.2 | 31.4 | 20.2 | 0.4 | 0.3 | 0.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.3 | 0.4 |
| C ₈ | 263.6 | 1592.7 | 88.7 | 710.8 | 41.8 | 4.2 | 4.2 | 5.4 | 3.6 | 11.7 | 3.2 | 9.8 | 3.3 | 17.1 | 6.5 |
| C ₉ | 2.3 | 54.6 | 20.7 | 13.2 | 0.5 | 1.7 | 0.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| C ₁₀ | 124.1 | 913.5 | 50.1 | 1094.0 | 15.4 | 1.1 | 1.1 | 14.0 | 0.5 | 1.4 | 0.1 | 1.2 | 0.2 | 2.9 | 0.4 |
| C ₁₁ | 0.2 | 12.7 | 8.4 | 8.4 | 2.1 | 1.3 | 0.7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.0 |
| C ₁₂ | 13.5 | 156.7 | 6.5 | 174.1 | 20.9 | 4.4 | 0.1 | 12.2 | 0.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.2 | 0.0 |
| C ₁₃ | 5.8 | 91.4 | 15.4 | 38.8 | 4.4 | 1.1 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.1 |
| C ₁₄ | 130.4 | 1354.2 | 56.7 | 449.0 | 20.1 | 12.4 | 0.7 | 0.7 | 0.0 | 1.8 | 1.4 | 1.6 | 1.4 | 3.2 | 1.0 |
| C ₁₅ | 1.5 | 32.3 | 8.1 | 16.2 | 1.8 | 0.8 | 0.7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| C ₁₆ | 50.9 | 583.9 | 20.9 | 416.2 | 19.6 | 9.8 | 0.3 | 0.2 | 0.2 | 0.0 | 0.1 | 0.0 | 0.1 | 4.3 | 0.0 |
| Média | 114.9 | 552.1 | 32.2 | 456.3 | 14.7 | 13.6 | 11.8 | 14.7 | 11.4 | 16.2 | 6.2 | 15.7 | 6.6 | 21.5 | 7.0 |

Tabela A.7: Comprimento médio da perda nos objetos não padronizados

| | COLA | Construtiva | | | | Residual | | | | | | | | | |
|-----------------|------|-------------|------------------|--------------|---------------------|----------|------------------|--------|---------------------|-------|--------------------|-------------|--------------------|-------|--------------------|
| | | FFD | FFD _A | Gulosa | Gulosa _A | FFD | FFD _A | Gulosa | Gulosa _A | RAG 1 | RAG _A 1 | RAG 2 | RAG _A 2 | RAG 3 | RAG _A 3 |
| C ₁ | 0.2 | 13.1 | 4.9 | 69.4 | 7.0 | 26.2 | 5.1 | 20.4 | 4.3 | 12.5 | 7.7 | 29.9 | 6.7 | 46.9 | 6.2 |
| C ₂ | 85.6 | 67.3 | 5.6 | 231.3 | 5.7 | 115.8 | 3.7 | 166.8 | 4.7 | 120.9 | 5.8 | 170.3 | 9.5 | 104.3 | 6.4 |
| C ₃ | 0.0 | 4.9 | 2.6 | 97.4 | 5.2 | 8.1 | 3.7 | 47.1 | 2.6 | 28.6 | 3.2 | 31.5 | 3.3 | 21.9 | 3.4 |
| C ₄ | 18.3 | 52.0 | 12.3 | 453.1 | 7.7 | 115.0 | 10.0 | 186.0 | 3.7 | 107.0 | 7.3 | 96.1 | 7.2 | 92.0 | 8.7 |
| C ₅ | 0.0 | 0.6 | 0.8 | 121.9 | 5.6 | 3.6 | 1.5 | 64.0 | 4.2 | 25.9 | 3.5 | 32.5 | 3.5 | 26.0 | 1.7 |
| C ₆ | 0.0 | 17.8 | 7.3 | 216.5 | 43.3 | 17.1 | 5.0 | 128.3 | 10.5 | 65.0 | 7.3 | 74.5 | 7.3 | 104.2 | 6.1 |
| C ₇ | 0.75 | 6.7 | 0.3 | 34.9 | 10.6 | 12.2 | 1.6 | 8.9 | 1.4 | 18.4 | 2.4 | 19.1 | 2.4 | 12.0 | 3.0 |
| C ₈ | 33.4 | 38.2 | 13.8 | 84.9 | 12.5 | 66.4 | 5.2 | 58.1 | 7.4 | 58.0 | 16.2 | 69.4 | 13.7 | 67.9 | 14.9 |
| C ₉ | 0.0 | 2.5 | 0.9 | 29.1 | 10.6 | 8.3 | 2.9 | 17.3 | 3.3 | 12.3 | 1.2 | 13.7 | 1.2 | 9.6 | 1.9 |
| C ₁₀ | 12.8 | 38.4 | 6.3 | 30.2 | 4.7 | 94.0 | 6.1 | 44.6 | 4.0 | 95.3 | 8.0 | 97.6 | 7.2 | 81.1 | 8.4 |
| C ₁₁ | 0.0 | 0.3 | 1.1 | 36.3 | 9.6 | 2.3 | 0.8 | 20.5 | 2.5 | 1.6 | 1.6 | 10.1 | 1.7 | 22.2 | 3.9 |
| C ₁₂ | 0.0 | 12.8 | 5.4 | 120.0 | 16.9 | 22.0 | 2.3 | 84.1 | 6.2 | 39.5 | 4.0 | 36.3 | 3.8 | 45.1 | 2.8 |
| C ₁₃ | 0.0 | 6.7 | 1.7 | 37.2 | 12.0 | 7.1 | 1.3 | 15.7 | 3.1 | 8.3 | 1.1 | 8.4 | 1.0 | 10.3 | 1.1 |
| C ₁₄ | 24.6 | 106.2 | 6.4 | 144.4 | 29.5 | 39.7 | 5.9 | 42.9 | 7.7 | 49.1 | 10.1 | 44.6 | 10.6 | 51.8 | 6.1 |
| C ₁₅ | 0.0 | 4.3 | 1.4 | 25.9 | 19.1 | 3.5 | 0.4 | 9.7 | 1.9 | 9.1 | 1.4 | 11.2 | 1.3 | 10.6 | 2.0 |
| C ₁₆ | 7.3 | 3.5 | 8.2 | 175.8 | 31.0 | 52.0 | 5.8 | 33.6 | 11.1 | 43.4 | 4.2 | 48.5 | 4.2 | 40.1 | 5.6 |
| Média | 11.4 | 23.5 | 4.9 | 119.3 | 14.4 | 37.1 | 3.8 | 59.3 | 4.9 | 43.4 | 5.3 | 49.6 | 5.3 | 46.6 | 5.1 |

Tabela A.8: Retalho Médio

| | Construtiva | | | | | Residual | | | | | | | | | |
|-----------------|-------------|--------------|------------------------|---------------|---------------------------|------------|------------------------|---------------|---------------------------|--------------|--------------------------|--------------|--------------------------|--------------|--------------------------|
| | <i>COLA</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> | <i>RAG 2</i> | <i>RAG_A 2</i> | <i>RAG 3</i> | <i>RAG_A 3</i> |
| C ₁ | 479.4 | 566.1 | 432.9 | 592.7 | 1161.9 | 465.2 | 513.9 | 221.7 | 385.3 | <i>94.0</i> | 359.6 | 105.2 | 409.8 | 136.9 | 375.1 |
| C ₂ | 2437.8 | 324.6 | 11806.8 | 1132.9 | 19668.5 | 268.0 | 776.4 | 230.5 | 1122.8 | 32.2 | 3791.9 | <i>0.0</i> | 3307.6 | 17.5 | 3664.7 |
| C ₃ | 536.9 | 490.3 | 536.9 | 414.6 | 1027.0 | 469.6 | 380.3 | 160.1 | 331.2 | <i>28.9</i> | 319.1 | 59.1 | 312.7 | 68.0 | 322.2 |
| C ₄ | 858.3 | 479.5 | 2298.4 | 1061.9 | 14682.2 | 44.4 | 612.3 | <i>0.0</i> | 853.0 | <i>0.0</i> | 1668.8 | <i>0.0</i> | 1460.8 | <i>0.0</i> | 1666.5 |
| C ₅ | 530.0 | 508.3 | 609.2 | 259.1 | 599.4 | 449.7 | 357.4 | 53.4 | 249.3 | <i>28.9</i> | 205.3 | 24.8 | 208.0 | 44.7 | 197.0 |
| C ₆ | 515.8 | 492.1 | 918.3 | 576.7 | 6021.0 | 390.6 | 581.7 | 69.9 | 722.9 | 94.3 | 420.1 | 92.4 | 420.1 | <i>83.4</i> | 632.6 |
| C ₇ | 577.1 | 393.2 | 591.5 | 417.0 | 1020.7 | 245.3 | 389.0 | 236.4 | 294.9 | <i>38.5</i> | 259.9 | 40.9 | 254.8 | 50.1 | 232.6 |
| C ₈ | 633.8 | 523.9 | 2939.8 | 1620.7 | 5621.0 | 251.6 | 476.9 | 401.8 | 549.7 | <i>39.1</i> | 652.4 | <i>39.1</i> | 606.0 | 70.4 | 540.6 |
| C ₉ | 410.2 | 416.4 | 520.4 | 332.3 | 662.9 | 262.7 | 206.4 | 130.7 | 333.2 | 42.7 | 153.2 | 51.7 | 154.8 | <i>31.5</i> | 179.1 |
| C ₁₀ | 568.6 | 573.8 | 961.0 | 1381.0 | 5000.1 | 239.3 | 442.1 | 86.9 | 511.0 | 7.7 | 304.4 | <i>0.0</i> | 347.3 | 34.4 | 333.3 |
| C ₁₁ | 571.0 | 520.3 | 485.4 | 231.5 | 358.3 | 312.8 | 353.1 | 58.1 | 364.8 | 30.9 | 200.5 | 39.1 | 233.8 | <i>11.2</i> | 223.1 |
| C ₁₂ | 617.3 | 550.2 | 426.8 | 493.6 | 2072.1 | 295.3 | 267.4 | 148.0 | 439.9 | <i>44.8</i> | 174.0 | 46.6 | 183.6 | 98.5 | 320.7 |
| C ₁₃ | 500.5 | 439.0 | 490.3 | 195.6 | 925.0 | 331.9 | 250.5 | 122.7 | 355.2 | <i>17.7</i> | 226.1 | 30.0 | 227.6 | 42.1 | 130.6 |
| C ₁₄ | 628.0 | 612.3 | 1341.4 | 437.8 | 4440.7 | 149.6 | 293.7 | 175.5 | 367.6 | <i>11.1</i> | 196.5 | 24.8 | 248.3 | 28.8 | 381.5 |
| C ₁₅ | 461.2 | 527.4 | 527.0 | 374.0 | 789.3 | 262.5 | 300.6 | 163.4 | 192.9 | <i>35.9</i> | 136.4 | 45.7 | 140.8 | 43.6 | 204.2 |
| C ₁₆ | 647.2 | 416.0 | 595.9 | 917.8 | 4262.8 | 251.8 | 225.2 | 1004.2 | 209.6 | <i>11.8</i> | 329.5 | 25.1 | 329.5 | 32.2 | 225.8 |
| Média | 685.8 | 489.6 | 1592.6 | 652.5 | 4269.6 | 293.1 | 401.7 | 204.0 | 455.2 | <i>34.9</i> | 587.4 | 39.0 | 552.8 | 49.6 | 601.9 |

Tabela A.9: Comprimento médio dos retalhos nos objetos padronizados

| | COLA | Construtiva | | | | Residual | | | | | | | | | |
|-----------------|--------------|-------------|------------------|--------|---------------------|----------|------------------|--------|---------------------|-------|--------------------|-------|--------------------|-------|--------------------|
| | | FFD | FFD _A | Gulosa | Gulosa _A | FFD | FFD _A | Gulosa | Gulosa _A | RAG 1 | RAG _A 1 | RAG 2 | RAG _A 2 | RAG 3 | RAG _A 3 |
| C ₁ | 479.4 | 566.1 | 419.8 | 555.3 | 889.4 | 460.3 | 381.7 | 206.4 | 244.8 | 38.7 | 183.6 | 39.9 | 222.8 | 46.9 | 233.6 |
| C ₂ | 2419.1 | 324.6 | 11709.6 | 1132.9 | 19129.3 | 268.0 | 404.1 | 230.5 | 542.4 | 32.2 | 3434.8 | 0.0 | 2999.9 | 17.5 | 3277.4 |
| C ₃ | 536.6 | 490.3 | 500.9 | 347.2 | 630.2 | 458.0 | 320.2 | 141.1 | 182.6 | 0.0 | 150.4 | 0.0 | 134.3 | 0.0 | 154.1 |
| C ₄ | 353.2 | 479.5 | 2144.5 | 767.5 | 14090.9 | 18.9 | 285.0 | 0.0 | 607.0 | 0.0 | 1406.2 | 0.0 | 1160.6 | 0.0 | 1362.3 |
| C ₅ | 531.0 | 508.3 | 593.3 | 220.4 | 320.2 | 446.5 | 298.1 | 40.5 | 85.8 | 0.0 | 47.0 | 0.0 | 47.0 | 0.0 | 66.7 |
| C ₆ | 514.5 | 492.1 | 749.2 | 395.4 | 5988.7 | 367.8 | 207.4 | 47.8 | 264.2 | 0.0 | 61.9 | 0.0 | 61.9 | 33.9 | 230.4 |
| C ₇ | 577.1 | 391.5 | 575.0 | 415.3 | 616.9 | 226.7 | 335.6 | 216.9 | 207.4 | 0.0 | 160.7 | 0.0 | 155.6 | 0.0 | 134.4 |
| C ₈ | 795.0 | 516.0 | 2550.1 | 1045.8 | 4509.3 | 188.6 | 102.6 | 110.7 | 117.8 | 0.0 | 261.5 | 0.0 | 223.5 | 18.8 | 243.6 |
| C ₉ | 410.0 | 416.4 | 517.0 | 315.2 | 395.9 | 246.6 | 160.4 | 101.2 | 224.2 | 0.0 | 48.8 | 0.0 | 48.8 | 0.0 | 85.7 |
| C ₁₀ | 565.6 | 573.8 | 876.3 | 672.8 | 3416.7 | 223.3 | 78.2 | 0.0 | 0.0 | 0.0 | 19.4 | 0.0 | 7.7 | 11.3 | 11.3 |
| C ₁₁ | 571.0 | 518.9 | 483.5 | 227.2 | 150.8 | 301.4 | 332.1 | 47.8 | 231.4 | 0.0 | 98.7 | 0.0 | 145.8 | 0.0 | 86.9 |
| C ₁₂ | 614.6 | 550.2 | 227.6 | 201.5 | 1157.2 | 213.8 | 17.9 | 53.7 | 52.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 54.1 |
| C ₁₃ | 497.7 | 439.0 | 486.0 | 146.3 | 647.7 | 309.8 | 217.6 | 107.9 | 287.1 | 0.0 | 172.0 | 0.0 | 173.3 | 0.0 | 43.2 |
| C ₁₄ | 628.0 | 606.6 | 1058.4 | 412.4 | 3300.7 | 94.9 | 120.4 | 125.2 | 132.4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 65.1 |
| C ₁₅ | 461.2 | 527.4 | 520.3 | 325.0 | 613.5 | 245.0 | 272.9 | 141.4 | 140.0 | 0.0 | 51.0 | 0.0 | 51.0 | 0.0 | 91.7 |
| C ₁₆ | 659.3 | 416.0 | 415.0 | 694.5 | 3150.4 | 217.3 | 29.6 | 167.2 | 0.0 | 0.0 | 40.2 | 0.0 | 40.2 | 0.0 | 0.0 |
| Média | 663.3 | 488.5 | 1489.2 | 492.2 | 3688.0 | 267.9 | 222.7 | 108.6 | 207.5 | 4.4 | 383.5 | 2.5 | 342.0 | 8.0 | 383.8 |

Tabela A.10: Comprimento médio dos retalhos nos objetos não padronizados

| | Construtiva | | | | | Residual | | | | | | | | | |
|-----------------|-------------|------------|------------------------|---------------|---------------------------|------------|------------------------|---------------|---------------------------|--------------|--------------------------|--------------|--------------------------|--------------|--------------------------|
| | <i>COLA</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>FFD</i> | <i>FFD_A</i> | <i>Gulosa</i> | <i>Gulosa_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> | <i>RAG 2</i> | <i>RAG_A 2</i> | <i>RAG 3</i> | <i>RAG_A 3</i> |
| C ₁ | 0.0 | 0.0 | 13.1 | 37.4 | 272.5 | 5.0 | 132.2 | 15.3 | 140.5 | 55.3 | 46.1 | 65.4 | 187.0 | 90.1 | 141.6 |
| C ₂ | 0.0 | 0.0 | 97.3 | 0.0 | 539.2 | 0.0 | 372.3 | 0.0 | 580.4 | 0.0 | 357.2 | 0.0 | 307.7 | 0.0 | 387.4 |
| C ₃ | 0.0 | 0.0 | 36.0 | 67.5 | 396.8 | 11.6 | 60.1 | 19.1 | 148.6 | 28.9 | 168.7 | 59.1 | 178.5 | 68.0 | 158.2 |
| C ₄ | 0.0 | 0.0 | 153.9 | 294.4 | 591.3 | 25.6 | 327.6 | 0.0 | 246.1 | 0.0 | 262.6 | 0.0 | 300.3 | 0.0 | 304.2 |
| C ₅ | 0.0 | 0.0 | 15.9 | 38.8 | 279.2 | 3.3 | 59.3 | 12.9 | 163.5 | 28.9 | 158.4 | 24.8 | 161.1 | 44.7 | 130.3 |
| C ₆ | 0.0 | 0.0 | 169.1 | 181.3 | 32.3 | 22.8 | 374.4 | 22.1 | 458.7 | 94.3 | 358.2 | 92.4 | 358.2 | 49.5 | 402.3 |
| C ₇ | 0.0 | 1.7 | 16.5 | 1.7 | 403.8 | 18.6 | 53.4 | 19.5 | 87.5 | 38.5 | 99.3 | 40.9 | 99.3 | 50.1 | 98.3 |
| C ₈ | 0.0 | 7.9 | 389.8 | 554.9 | 1111.7 | 63.0 | 374.4 | 291.1 | 432.0 | 39.1 | 390.9 | 39.1 | 382.6 | 51.6 | 297.1 |
| C ₉ | 0.0 | 0.0 | 3.4 | 17.1 | 267.0 | 16.1 | 46.1 | 29.5 | 109.0 | 42.7 | 104.4 | 51.7 | 106.0 | 31.5 | 93.4 |
| C ₁₀ | 0.0 | 0.0 | 84.7 | 708.2 | 1583.5 | 16.0 | 364.0 | 86.9 | 511.0 | 7.7 | 285.0 | 0.0 | 339.6 | 23.1 | 322.0 |
| C ₁₁ | 0.0 | 2.4 | 1.9 | 4.3 | 207.5 | 11.4 | 21.0 | 9.5 | 133.4 | 30.9 | 101.9 | 39.1 | 88.0 | 11.2 | 136.2 |
| C ₁₂ | 0.0 | 0.0 | 199.3 | 293.1 | 914.9 | 81.2 | 249.5 | 94.3 | 387.1 | 44.8 | 174.0 | 46.6 | 183.6 | 98.5 | 266.6 |
| C ₁₃ | 0.0 | 0.0 | 4.1 | 49.3 | 277.3 | 22.1 | 32.9 | 14.8 | 88.1 | 17.7 | 54.0 | 30.0 | 54.4 | 42.1 | 87.4 |
| C ₁₄ | 0.0 | 5.8 | 283.0 | 25.5 | 1140.0 | 54.8 | 173.4 | 50.3 | 235.3 | 11.1 | 196.5 | 24.8 | 248.3 | 28.8 | 316.5 |
| C ₁₅ | 0.0 | 0.0 | 6.7 | 48.9 | 174.8 | 17.5 | 25.7 | 22.0 | 52.9 | 35.9 | 85.4 | 45.7 | 89.9 | 43.6 | 112.5 |
| C ₁₆ | 0.0 | 0.0 | 180.9 | 223.3 | 1112.4 | 34.5 | 195.6 | 837.0 | 209.6 | 11.8 | 289.3 | 25.1 | 289.3 | 32.2 | 225.8 |
| Média | 0.0 | 1.1 | 103.5 | 159.1 | 581.5 | 25.2 | 178.9 | 95.3 | 249.0 | 30.5 | 195.7 | 36.5 | 210.9 | 41.6 | 217.5 |

Tabela A.11: Número médio de objetos cortados com retalho

| | COLA | Construtiva | | | | Residual | | | | | | | | | |
|-----------------|------|-------------|------------------|--------|---------------------|----------|------------------|--------|---------------------|-------|--------------------|-------|--------------------|-------|--------------------|
| | | FFD | FFD _A | Gulosa | Gulosa _A | FFD | FFD _A | Gulosa | Gulosa _A | RAG 1 | RAG _A 1 | RAG 2 | RAG _A 2 | RAG 3 | RAG _A 3 |
| C ₁ | 3.0 | 1.0 | 1.6 | 2.7 | 10.1 | 0.8 | 2.5 | 0.6 | 2.2 | 0.6 | 2.2 | 0.6 | 2.3 | 1.0 | 2.0 |
| C ₂ | 13.1 | 0.6 | 36.0 | 4.2 | 53.6 | 0.4 | 2.4 | 0.4 | 3.4 | 0.1 | 11.3 | 0.0 | 9.3 | 0.1 | 10.8 |
| C ₃ | 2.2 | 1.0 | 1.6 | 2.1 | 8.6 | 0.9 | 1.5 | 0.5 | 2.0 | 0.3 | 2.2 | 0.5 | 2.2 | 0.8 | 2.5 |
| C ₄ | 4.3 | 0.8 | 8.0 | 4.1 | 36.8 | 0.2 | 1.9 | 0.0 | 2.3 | 0.0 | 4.9 | 0.0 | 4.8 | 0.0 | 4.7 |
| C ₅ | 1.1 | 1.0 | 1.2 | 1.6 | 4.8 | 0.9 | 1.5 | 0.3 | 2.1 | 0.5 | 2.1 | 0.3 | 2.2 | 0.6 | 2.0 |
| C ₆ | 5.1 | 0.9 | 2.9 | 1.8 | 17.8 | 0.9 | 2.1 | 0.2 | 2.6 | 0.4 | 1.6 | 0.4 | 1.6 | 0.3 | 2.1 |
| C ₇ | 2.0 | 0.9 | 1.5 | 3.9 | 11.8 | 0.8 | 1.8 | 0.6 | 2.0 | 0.5 | 2.1 | 0.6 | 2.1 | 0.6 | 1.6 |
| C ₈ | 4.2 | 0.9 | 16.7 | 9.0 | 27.5 | 0.7 | 2.8 | 2.0 | 2.8 | 0.3 | 3.1 | 0.3 | 3.0 | 0.5 | 3.3 |
| C ₉ | 1.2 | 1.0 | 1.1 | 2.6 | 7.8 | 0.8 | 1.3 | 0.6 | 2.8 | 0.6 | 1.5 | 0.6 | 1.5 | 0.6 | 1.8 |
| C ₁₀ | 2.4 | 0.9 | 3.1 | 7.2 | 26.6 | 0.4 | 2.0 | 0.6 | 2.8 | 0.1 | 1.5 | 0.0 | 1.7 | 0.2 | 1.6 |
| C ₁₁ | 1.1 | 1.0 | 1.0 | 0.6 | 5.4 | 0.9 | 1.1 | 0.4 | 3.7 | 0.5 | 1.8 | 0.6 | 1.7 | 0.2 | 2.6 |
| C ₁₂ | 5.3 | 0.9 | 1.8 | 2.8 | 9.3 | 0.8 | 1.6 | 1.0 | 2.4 | 0.3 | 1.1 | 0.3 | 1.2 | 0.6 | 1.7 |
| C ₁₃ | 3.2 | 1.0 | 1.6 | 3.2 | 11.4 | 0.8 | 1.5 | 0.5 | 2.6 | 0.3 | 1.7 | 0.4 | 1.7 | 0.6 | 1.5 |
| C ₁₄ | 3.0 | 1.4 | 7.4 | 1.3 | 22.7 | 0.6 | 1.7 | 0.7 | 2.3 | 0.1 | 1.8 | 0.2 | 2.1 | 0.3 | 2.8 |
| C ₁₅ | 1.2 | 1.0 | 1.1 | 3.0 | 7.6 | 0.8 | 1.1 | 0.6 | 1.6 | 0.5 | 1.3 | 0.5 | 1.4 | 0.6 | 2.2 |
| C ₁₆ | 2.5 | 0.9 | 2.3 | 4.1 | 20.5 | 0.5 | 1.7 | 5.9 | 1.6 | 0.1 | 2.1 | 0.2 | 2.1 | 0.3 | 1.5 |
| Média | 3.4 | 1.0 | 5.6 | 3.9 | 17.6 | 0.7 | 1.8 | 0.9 | 2.5 | 0.3 | 2.6 | 0.3 | 2.6 | 0.5 | 2.8 |

Tabela A.12: Número médio de objetos cortados com perda pequena

| | COLA | Construtiva | | | | Residual | | | | | | | | | |
|-----------------|------|-------------|------------------|--------|---------------------|----------|------------------|--------|---------------------|-------|--------------------|-------|--------------------|-------|--------------------|
| | | FFD | FFD _A | Gulosa | Gulosa _A | FFD | FFD _A | Gulosa | Gulosa _A | RAG 1 | RAG _A 1 | RAG 2 | RAG _A 2 | RAG 3 | RAG _A 3 |
| C ₁ | 3.35 | 68.8 | 13.0 | 6.1 | 6.2 | 5.8 | 2.2 | 2.1 | 2.2 | 1.3 | 2.1 | 1.5 | 2.0 | 1.5 | 2.3 |
| C ₂ | 23.8 | 34.8 | 38.8 | 14.2 | 14.5 | 25.1 | 25.2 | 24.2 | 24.3 | 24.2 | 26.2 | 23.7 | 26.5 | 25.0 | 26.2 |
| C ₃ | 1.75 | 56.7 | 7.1 | 4.2 | 4.4 | 5.1 | 1.2 | 1.3 | 1.3 | 1.1 | 1.0 | 1.0 | 1.1 | 0.8 | 1.4 |
| C ₄ | 18.3 | 48.6 | 27.8 | 11.0 | 11.5 | 11.9 | 9.1 | 7.6 | 7.7 | 8.2 | 7.6 | 8.1 | 7.2 | 8.8 | 8.9 |
| C ₅ | 0.4 | 32.1 | 3.4 | 2.1 | 2.1 | 3.7 | 1.2 | 0.8 | 0.9 | 1.0 | 1.0 | 1.0 | 0.9 | 0.8 | 0.8 |
| C ₆ | 5.4 | 64.9 | 12.0 | 15.3 | 17.0 | 7.1 | 2.0 | 2.0 | 2.2 | 0.8 | 1.2 | 1.0 | 1.2 | 1.5 | 1.4 |
| C ₇ | 3.2 | 50.8 | 17.0 | 2.3 | 2.5 | 3.7 | 1.1 | 0.6 | 0.6 | 1.0 | 0.9 | 1.0 | 0.9 | 1.2 | 1.4 |
| C ₈ | 25.6 | 39.9 | 37.2 | 16.7 | 17.2 | 7.7 | 3.6 | 2.8 | 2.9 | 3.1 | 3.2 | 3.2 | 3.0 | 4.3 | 4.8 |
| C ₉ | 1.0 | 38.1 | 10.7 | 2.6 | 3.0 | 3.3 | 1.5 | 1.0 | 1.1 | 0.6 | 0.6 | 0.6 | 0.6 | 0.9 | 0.8 |
| C ₁₀ | 12.6 | 48.4 | 25.0 | 7.2 | 7.3 | 7.7 | 2.1 | 1.1 | 1.2 | 1.2 | 1.5 | 1.2 | 1.4 | 1.1 | 1.5 |
| C ₁₁ | 0.2 | 11.6 | 4.6 | 3.5 | 4.0 | 1.6 | 1.0 | 1.0 | 1.0 | 0.5 | 0.6 | 0.6 | 0.7 | 1.1 | 1.4 |
| C ₁₂ | 5.2 | 64.4 | 4.9 | 11.8 | 11.9 | 4.8 | 0.7 | 1.7 | 1.7 | 0.7 | 0.9 | 0.7 | 0.8 | 0.7 | 0.7 |
| C ₁₃ | 2.8 | 50.4 | 11.2 | 5.2 | 5.5 | 3.2 | 0.8 | 1.6 | 1.7 | 0.7 | 0.5 | 0.5 | 0.4 | 0.8 | 0.7 |
| C ₁₄ | 22.7 | 37.0 | 31.6 | 15.6 | 16.1 | 6.9 | 2.0 | 1.9 | 1.9 | 2.4 | 2.4 | 2.4 | 2.3 | 1.9 | 3.1 |
| C ₁₅ | 0.8 | 27.4 | 5.8 | 4.7 | 5.5 | 1.1 | 0.7 | 1.0 | 1.0 | 0.7 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 |
| C ₁₆ | 11.7 | 57.4 | 15.2 | 13.7 | 13.7 | 7.9 | 1.9 | 2.2 | 2.3 | 1.4 | 1.3 | 1.3 | 1.3 | 1.3 | 1.3 |
| Média | 9.0 | 45.7 | 16.6 | 8.5 | 8.9 | 6.7 | 3.5 | 3.3 | 3.4 | 3.1 | 3.2 | 3.1 | 3.2 | 3.3 | 3.6 |

Tabela A.13: Número médio de objetos cortados com perda não tão pequena

| | COLA | Construtiva | | | | Residual | | | | | | | | | |
|-----------------|------|-------------|------------------|------------|---------------------|----------|------------------|--------|---------------------|-------|--------------------|-------|--------------------|-------|--------------------|
| | | FFD | FFD _A | Gulosa | Gulosa _A | FFD | FFD _A | Gulosa | Gulosa _A | RAG 1 | RAG _A 1 | RAG 2 | RAG _A 2 | RAG 3 | RAG _A 3 |
| C ₁ | 1.4 | 16.5 | 0.0 | 6.4 | 0.0 | 1.6 | 0.0 | 1.3 | 0.0 | 0.6 | 0.0 | 1.4 | 0.0 | 0.9 | 0.0 |
| C ₂ | 29.3 | 77.2 | 0.0 | 36.6 | 0.0 | 11.1 | 7.8 | 10.2 | 7.8 | 10.6 | 1.0 | 11.0 | 1.9 | 10.8 | 2.3 |
| C ₃ | 0.1 | 1.0 | 0.0 | 6.0 | 0.0 | 0.3 | 0.0 | 1.7 | 0.0 | 1.3 | 0.0 | 1.7 | 0.0 | 1.0 | 0.0 |
| C ₄ | 14.1 | 47.5 | 0.0 | 26.2 | 0.0 | 4.9 | 1.2 | 3.8 | 1.2 | 2.9 | 0.0 | 2.7 | 0.0 | 3.5 | 0.0 |
| C ₅ | 0.0 | 0.1 | 0.0 | 4.5 | 0.0 | 0.1 | 0.0 | 2.7 | 0.0 | 1.1 | 0.0 | 1.5 | 0.0 | 1.2 | 0.0 |
| C ₆ | 2.1 | 22.9 | 0.0 | 16.0 | 0.0 | 1.9 | 0.0 | 2.4 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 1.5 | 0.0 |
| C ₇ | 0.0 | 2.7 | 0.0 | 4.9 | 0.0 | 1.1 | 0.0 | 1.1 | 0.0 | 1.6 | 0.0 | 2.0 | 0.0 | 0.7 | 0.0 |
| C ₈ | 9.7 | 60.9 | 0.0 | 15.3 | 0.0 | 1.7 | 0.0 | 1.1 | 0.0 | 1.1 | 0.0 | 1.1 | 0.0 | 2.2 | 0.0 |
| C ₉ | 0.1 | 0.1 | 0.0 | 3.9 | 0.0 | 0.3 | 0.0 | 2.2 | 0.0 | 1.1 | 0.0 | 1.2 | 0.0 | 0.7 | 0.0 |
| C ₁₀ | 3.8 | 42.9 | 0.0 | 15.1 | 0.0 | 2.6 | 0.0 | 1.1 | 0.0 | 1.7 | 0.0 | 1.8 | 0.0 | 1.2 | 0.0 |
| C ₁₁ | 0.0 | 0.0 | 0.0 | 3.9 | 0.0 | 0.1 | 0.0 | 2.2 | 0.0 | 0.9 | 0.0 | 0.8 | 0.0 | 1.5 | 0.0 |
| C ₁₂ | 0.4 | 3.9 | 0.0 | 4.3 | 0.0 | 0.3 | 0.0 | 1.3 | 0.0 | 0.7 | 0.0 | 0.6 | 0.0 | 0.7 | 0.0 |
| C ₁₃ | 0.0 | 1.6 | 0.0 | 5.5 | 0.0 | 0.7 | 0.0 | 1.9 | 0.0 | 1.0 | 0.0 | 1.3 | 0.0 | 1.4 | 0.0 |
| C ₁₄ | 5.3 | 53.5 | 0.0 | 17.0 | 0.0 | 0.9 | 0.0 | 1.2 | 0.0 | 1.2 | 0.0 | 1.1 | 0.0 | 1.7 | 0.0 |
| C ₁₅ | 0.0 | 0.5 | 0.0 | 3.1 | 0.0 | 0.2 | 0.0 | 0.9 | 0.0 | 0.9 | 0.0 | 1.0 | 0.0 | 1.2 | 0.0 |
| C ₁₆ | 1.7 | 28.7 | 0.0 | 14.7 | 0.0 | 2.0 | 0.0 | 0.7 | 0.0 | 1.1 | 0.0 | 1.1 | 0.0 | 1.0 | 0.0 |
| Média | 4.3 | 22.5 | 0.0 | 11.4 | 0.0 | 1.9 | 0.6 | 2.2 | 0.6 | 1.8 | 0.1 | 2.0 | 0.1 | 2.0 | 0.1 |

Experimentos da Seção 7.3

Nesta seção do Apêndice A apresentamos o resultado médio obtido por cada classe de exemplos geradas na Seção 7.3.

Tabela A.14: Média da perda relativa nos objetos padronizados

| | Construtiva | | | Residual | |
|-----------------|--------------------|------------|------------------------|-----------------|--------------------------|
| | <i>COLA</i> | <i>FFD</i> | <i>FFD_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> |
| C ₁ | 0,0016 | 0,0039 | 0,0003 | 0,0020 | 0,0007 |
| C ₂ | 0,0257 | 0,0237 | 0,0005 | 0,0085 | 0,0020 |
| C ₃ | 0,0115 | 0,0031 | 0,0001 | 0,0014 | 0,0001 |
| C ₄ | 0,0492 | 0,0103 | 0,0004 | 0,0029 | 0,0003 |
| C ₅ | 0,0037 | 0,0012 | 0,0000 | 0,0001 | 0,0000 |
| C ₆ | 0,0340 | 0,0066 | 0,0002 | 0,0007 | 0,0000 |
| C ₇ | 0,0000 | 0,0033 | 0,0001 | 0,0009 | 0,0003 |
| C ₈ | 0,0194 | 0,0269 | 0,0006 | 0,0082 | 0,0021 |
| C ₉ | 0,0021 | 0,0021 | 0,0001 | 0,0001 | 0,0000 |
| C ₁₀ | 0,0388 | 0,0119 | 0,0004 | 0,0020 | 0,0004 |
| C ₁₁ | 0,0114 | 0,0018 | 0,0000 | 0,0000 | 0,0000 |
| C ₁₂ | 0,0382 | 0,0091 | 0,0002 | 0,0009 | 0,0000 |
| C ₁₃ | 0,0000 | 0,0056 | 0,0001 | 0,0002 | 0,0002 |
| C ₁₄ | 0,0066 | 0,0217 | 0,0004 | 0,0077 | 0,0024 |
| C ₁₅ | 0,0005 | 0,0021 | 0,0000 | 0,0001 | 0,0001 |
| C ₁₆ | 0,0214 | 0,0163 | 0,0002 | 0,0013 | 0,0002 |
| C ₁₇ | 0,0131 | 0,0014 | 0,0000 | 0,0000 | 0,0000 |
| C ₁₈ | 0,0531 | 0,0067 | 0,0002 | 0,0004 | 0,0000 |

Tabela A.15: Média da perda relativa nos objetos não padronizados

| | Construtiva | | | Residual | |
|-----------------|-------------|------------|------------------------|--------------|--------------------------|
| | <i>COLA</i> | <i>FFD</i> | <i>FFD_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> |
| C ₁ | 0,0810 | 0,0517 | 0,0053 | 0,0933 | 0,0048 |
| C ₂ | 0,0954 | 0,0736 | 0,0102 | 0,1000 | 0,0108 |
| C ₃ | 0,0252 | 0,0349 | 0,0047 | 0,0872 | 0,0082 |
| C ₄ | 0,0619 | 0,0450 | 0,0165 | 0,0909 | 0,0093 |
| C ₅ | 0,0008 | 0,0082 | 0,0034 | 0,0914 | 0,0052 |
| C ₆ | 0,0074 | 0,0190 | 0,0157 | 0,1409 | 0,0090 |
| C ₇ | 0,0953 | 0,0531 | 0,0086 | 0,1045 | 0,0064 |
| C ₈ | 0,0781 | 0,0648 | 0,0122 | 0,1108 | 0,0100 |
| C ₉ | 0,0555 | 0,0301 | 0,0052 | 0,0921 | 0,0040 |
| C ₁₀ | 0,0830 | 0,0369 | 0,0136 | 0,0933 | 0,0077 |
| C ₁₁ | 0,0056 | 0,0100 | 0,0025 | 0,0778 | 0,0061 |
| C ₁₂ | 0,0218 | 0,0164 | 0,0136 | 0,1418 | 0,0054 |
| C ₁₃ | 0,1122 | 0,0338 | 0,0058 | 0,0975 | 0,0060 |
| C ₁₄ | 0,0559 | 0,0447 | 0,0136 | 0,0586 | 0,0086 |
| C ₁₅ | 0,0805 | 0,0157 | 0,0052 | 0,0725 | 0,0039 |
| C ₁₆ | 0,0832 | 0,0275 | 0,0129 | 0,1227 | 0,0077 |
| C ₁₇ | 0,0162 | 0,0081 | 0,0024 | 0,0734 | 0,0050 |
| C ₁₈ | 0,0417 | 0,0166 | 0,0107 | 0,1108 | 0,0072 |
| Média | 0,0556 | 0,0328 | 0,0090 | 0,0977 | 0,0070 |

Tabela A. 16: Média do retalho líquido relativo

| | Construtiva | | | Residual | |
|-----------------|-------------|------------|------------------------|--------------|--------------------------|
| | <i>COLA</i> | <i>FFD</i> | <i>FFD_A</i> | <i>RAG 1</i> | <i>RAG_A 1</i> |
| C ₁ | 0,3448 | 0,6818 | 0,8949 | 0,8627 | 0,9061 |
| C ₂ | 0,3347 | 0,6060 | 1,1075 | 0,9023 | 1,0032 |
| C ₃ | 0,0836 | 0,6086 | 0,8555 | 0,8872 | 0,9029 |
| C ₄ | 0,1226 | 0,5089 | 1,0907 | 0,9193 | 0,9971 |
| C ₅ | 0,0637 | 0,5119 | 0,8219 | 0,8691 | 0,9011 |
| C ₆ | 0,0703 | 0,4291 | 1,1449 | 0,8851 | 0,9723 |
| C ₇ | 0,4769 | 0,7662 | 0,9009 | 0,9130 | 0,9333 |
| C ₈ | 0,5583 | 0,7373 | 1,0311 | 0,9227 | 0,9801 |
| C ₉ | 0,2199 | 0,7220 | 0,9037 | 0,9112 | 0,9393 |
| C ₁₀ | 0,1949 | 0,5775 | 1,0005 | 0,9160 | 0,9739 |
| C ₁₁ | 0,0375 | 0,6214 | 0,8377 | 0,9115 | 0,9405 |
| C ₁₂ | 0,0687 | 0,5368 | 1,0005 | 0,9190 | 0,9696 |
| C ₁₃ | 0,5913 | 0,8246 | 0,9171 | 0,9243 | 0,9439 |
| C ₁₄ | 0,6058 | 0,7840 | 0,9650 | 0,9190 | 0,9451 |
| C ₁₅ | 0,3590 | 0,7669 | 0,9106 | 0,9313 | 0,9446 |
| C ₁₆ | 0,3068 | 0,6367 | 0,9541 | 0,9234 | 0,9493 |
| C ₁₇ | 0,0631 | 0,6830 | 0,8838 | 0,9342 | 0,9481 |
| C ₁₈ | 0,0693 | 0,6019 | 0,9511 | 0,9468 | 0,9650 |
| Média | 0,2540 | 0,6447 | 0,9540 | 0,9110 | 0,9508 |