



## GERAÇÃO DE PADRÕES DE CORTE BIDIMENSIONAIS GUILHOTINADOS VIA GRASP

**André Soares Velasco**

Instituto Federal Fluminense - IFF/ Universidade Federal Fluminense - UFF  
Av. Souza Mota, 350. Parque Fundão - Campos dos Goytacazes - RJ. CEP: 28060-010  
asvelasco@iff.edu.br

**Eduardo Uchoa Barboza**

Dep. de Engenharia de Produção - Universidade Federal Fluminense - UFF  
Rua Passo da Pátria 156, Bloco D. São Domingos - Niterói - RJ. CEP: 22210-240  
uchoa@producao.uff.br

### RESUMO

Este trabalho considera um caso particular dos problemas de corte, conhecido na literatura como Problema de Corte Bidimensional Guilhotinado e Restrito, que pertence à classe de problemas NP-Difícil. O trabalho investiga o uso da metaheurística GRASP com o objetivo de encontrar boas soluções, ou até mesmo o ótimo, em tempos de execução condizentes com a realidade. São propostos dois algoritmos para a geração de padrões de corte, com ampliações nas estratégias utilizadas pelo algoritmo conhecido como GRASP-2D. Os testes computacionais em instâncias da literatura indicam um bom desempenho dos algoritmos propostos.

**PALAVRAS CHAVE.** Geração de Padrões de Corte Bidimensionais. GRASP.

### ABSTRACT

This work considers a particular case of cutting stock problems known in the literature as Constrained Two-Dimensional Guillotine Cutting Stock Problem, that belongs to the class of NP-hard problems. The work investigates the use of the GRASP metaheuristics with the aim at finding good solutions or even the optimal within realistic execution times. Two algorithms are proposed for the cutting pattern generation, enhancing the strategies used by the algorithm known as GRASP-2D. The computational tests in instances from the literature show a good performance of the proposed algorithms.

**KEYWORDS.** Generation of Two-dimensional Cutting Patterns. GRASP.

## 1. Introdução

O planejamento operacional de ações que consistem na transformação de materiais em produtos semi-acabados ou finais, não somente evidenciam a tecnologia de processos como também destacam a produtividade, utilizando de forma inteligente os recursos da produção. Com o intuito de minimizar o desperdício em transformações promovidas pelo corte em madeiras, metais, vidros, entre outras matérias-primas, o presente trabalho aprecia o problema de corte bidimensional, cujas sobras praticamente não são reaproveitadas, admitindo que a melhoria e a eficiência desses processos ocasionariam estimável redução nos custos, agregando diferencial competitivo a esses produtos.

O Problema de Corte Bidimensional Guilhotinado e Restrito (PCBGR) é um caso particular dos Problemas de Corte e Empacotamento que consiste em determinar a melhor forma de se gerar uma quantidade de peças retangulares (itens), com demandas pré-estabelecidas, realizando cortes ortogonais do tipo guilhotinado, em placas retangulares em estoque (objetos), com quantidade suficiente para atender a demanda. Uma solução ótima para este problema de corte exhibe padrões bidimensionais, que atendam a uma demanda de itens e apresentem perda mínima de material em sua produção. Aos itens, têm-se associado um valor de utilidade que pode ser a medida de sua área ou estar relacionado à sua importância em presença aos outros itens.

Os padrões de corte são denominados bidimensionais, pois existem exclusivamente duas dimensões relevantes em suas composições, o comprimento e a largura. O corte é dito guilhotinado ortogonal se quando for efetuado, gerar dois outros retângulos. Caso haja limitação à quantidade de itens produzidos, o padrão de corte é considerado restrito. Sabendo que o tipo de corte e o número de mudanças permitidas na direção dos cortes (estágios) são questões importantes a serem consideradas na geração destes padrões, a figura 1 destaca os dois tipos de padrões resultantes de cortes ortogonais, com cinco itens em ambos, sendo o padrão de corte guilhotinado configurado com cinco estágios. Ainda nessa figura, os  $k$  estágios são indicados por  $G_k$  e consideram uma sequência de rotações de  $90^\circ$  na direção dos cortes, paralelas aos lados do objeto.

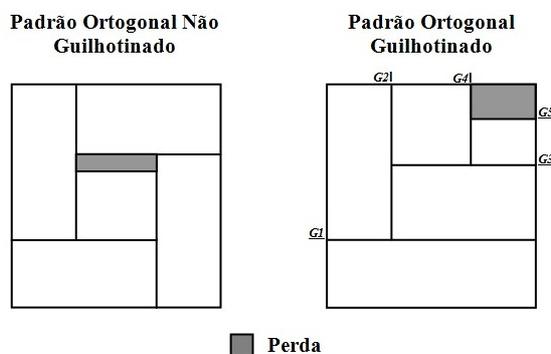


FIGURA 1 – Padrões de Corte Bidimensionais Ortogonais.

No problema de geração de padrões de corte, os itens não necessariamente são produzidos em sua totalidade e o objetivo é maximizar o valor de utilidade total dos itens gerados no padrão. Este problema pode ser considerado um subproblema do PCBGR, no qual se deseja selecionar padrões de corte e suas respectivas repetições, com a necessidade de atender a demanda de itens, de forma que o valor de utilidade total dos objetos seja minimizado.

Na tipologia para os Problemas de Corte e Empacotamento, apresentada por Dyckhoff (1990), estes problemas recebem classificação distintas. O PCBDR é indicado pela quádrupla ordenada  $2/V/I/R$ , que determina ser um problema da classe bidimensional (2), com todos os itens sendo designados a uma seleção de objetos (V), estes de formatos idênticos (I) e itens de formatos distintos em relativa quantidade (R). Enquanto, a geração de padrões para o PCBGR é apontada pela quádrupla  $2/B/O/R$ , onde (B) determina a alocação de todos os objetos e uma parte dos itens e (O) delimita o problema a um objeto.

Sendo a geração de padrões de corte para o PCBGR pertencente à classe de problemas *NP*-difícil (HIFI, 2004), a utilização de um método exato para obter o melhor padrão acarreta em esforço computacional considerável e torna-se inviável em situações práticas. Considerando esta dificuldade, a utilização de procedimentos heurísticos para geração destes padrões de corte ganha destaque na literatura e algumas destas abordagens podem ser apreciadas nas seguintes referências: Wang (1983), Oliveira e Ferreira (1990), Fayard *et al.* (1998), Amaral e Wright (2001), Alvarez-Valdés *et al.* (2002), Hifi (2004), Morabito e Pureza (2010).

Em decorrência, o presente trabalho enfatiza a utilização da metaheurística GRASP, *Greedy Randomized Adaptive Search Procedure*, apresentando as ampliações do algoritmo GRASP-2D (VELASCO *et al.*, 2008), em duas versões que se distinguem quanto ao valor de utilidade do item, na geração de padrões de corte para o PCBGR sem limite de estágios e sem rotação dos itens. Para analisar o desempenho dos algoritmos propostos, foram realizados testes computacionais em instâncias clássicas e os resultados comparados com os ótimos da literatura.

## 2. O problema de geração de padrões de corte

Seja um objeto *R* de comprimento *C* e largura *L*, denotado por  $R = (C, L)$  e *n* itens com demanda  $d_i$  e dimensões fixas  $p_i = (c_i, l_i)$ , com  $i = 1, \dots, n$ . O problema de geração de padrões para o PCBGR consiste em determinar os arranjos destes itens que não extrapolem as dimensões do objeto, nem apresentem superposição dos mesmos e respeitem os limites  $d_i$  por meio de cortes guilhotinados. Sendo assim, o problema de determinar um padrão de corte com maior valor de utilidade, depois de obtidos os itens, pode ser descrito pelo modelo analítico a seguir.

$$\begin{aligned} \text{Max } Z &= \sum_{i=1}^n v_i x_i \\ \text{s.a: } & [x_1, \dots, x_n] \\ & x_i \geq 0 \text{ e Inteiro} \end{aligned}$$

Onde:

- $x_i$  é a variável que representa número de vezes que o item *i* é obtido no padrão;
- $v_i$  é a constante que indica o valor de utilidade do item *i*;
- $[x_1, \dots, x_n]$  é o vetor que representa um padrão de corte bidimensional guilhotinado viável.

Diante da dificuldade de se apresentar uma formulação que utiliza cortes guilhotinados sem limite de estágios, também denominados não estagiados, para gerar um padrão para o PCBGR, são desenvolvidos dois algoritmos heurísticos, baseados na metodologia GRASP, a partir de ampliações feitas no algoritmo GRASP-2D, para a resolução do problema supracitado, com estratégias que se distinguem de acordo com a maneira como é determinado o valor de utilidade de cada item. A versão GRASP-2D<sub>A</sub> é utilizada quando o valor de utilidade  $v_i$ , associado a cada item *i*, é igual à medida de área deste item, isto é,  $v_i = c_i \cdot l_i$ ,  $i = 1, \dots, n$ . Se no conjunto dos itens demandados existe pelo menos um item *i* tal que  $v_i \neq c_i \cdot l_i$ , a versão utilizada é denominada GRASP-2D<sub>V</sub>.

## 3. GRASP

A metaheurística GRASP, *Greedy Randomized Adaptive Search Procedure*, foi proposta por Feo e Resende (1989) e é fundamentalmente uma combinação de uma heurística Construtiva e um procedimento de Busca Local. Este procedimento é iterativo probabilístico que atua por amostragem em um espaço de soluções subtraído. A cada iteração é gerada uma solução de forma míope, aleatória e adaptativa que será condicionada à Busca Local e a melhor solução encontrada é mantida como resultado final. Sendo o algoritmo GRASP composto a cada iteração de duas fases, a primeira de construção e a segunda de melhoria, estas são descritas separadamente.

### 3.1 Fase de Construção da GRASP

Basicamente, na primeira fase, uma solução viável é construída iterativamente apreciando um elemento de cada vez. Este elemento é retirado aleatoriamente de um conjunto

restrito denominado de lista restrita de candidatos (LRC). A LRC é composta pelos elementos mais interessantes de uma lista constituída de candidatos a serem incluídos em uma solução, seguindo um critério guloso. Esta técnica de escolha permite que soluções distintas sejam produzidas em cada iteração GRASP. A cada elemento incluído na solução, adaptações relacionadas às informações do problema são feitas sob a orientação de uma função gulosa. Esta função gulosa mede o benefício, associado à admissão de cada elemento a uma determinada solução.

Um parâmetro  $\alpha$  determina a quantidade de elementos incluídos no LRC, assumindo valores no intervalo  $[0,1]$ . Seja  $P$  um conjunto finito formado pelos próximos elementos  $p_i$  a serem incluídos na solução,  $f : P \rightarrow \mathbb{R}$  uma função gulosa e  $\beta = \max \{f(p_i); p_i \in P\}$  o melhor valor retornado pela função gulosa. A LRC é determinada aplicando o parâmetro  $\alpha$  ao valor  $\beta$  na definição de um intervalo de valores, isto é,  $LRC = \{p_i \in P / \alpha \cdot \beta \leq f(p_i) \leq \beta\}$ . Desta forma, para  $\alpha = 1$  se gera soluções totalmente gulosas e para  $\alpha = 0$  se produz soluções totalmente aleatórias.

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

### 3.2 Fase de Melhoria da GRASP

A segunda fase é tipicamente um procedimento de Busca Local e a rapidez com que o processo de busca local encontra uma solução melhor é diretamente proporcional à qualidade da solução gerada na fase de construção. Baseando-se na construção eficiente de uma solução inicial, pode-se conseguir uma redução no número de passos necessários para se alcançar uma solução desejável na fase de melhoria e, conseqüentemente, a diminuição do tempo computacional consumido no processamento desta solução.

### 4. Algoritmo GRASP-2D<sub>A</sub>

Inicialmente, o algoritmo GRASP-2D, proposto por Velasco (2005), foi implementado para resolver o PCBGR não estagiado e com rotação dos itens, com resultados bastante promissores em testes realizados em instâncias de até vinte itens, quando comparado as heurísticas Busca Tabu e *Simulated Annealing*, demonstrando que a metodologia GRASP pode ser aplicada com sucesso ao problema em destaque (SILVA *et al.*, 2013). Sendo o algoritmo GRASP-2D<sub>A</sub> projetado a partir de ampliações no algoritmo supracitado, para geração de padrões do PCBGR não estagiado e sem rotação de itens, é indispensável uma breve apresentação das definições necessárias na arquitetura do GRASP-2D.

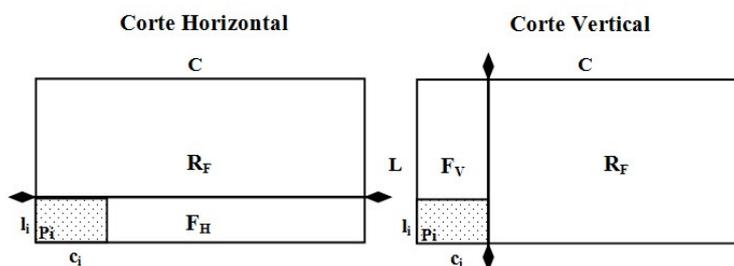


FIGURA 2 - Cortes guilhotinados e faixas guilhotinas.

Dado um estoque de objetos retangulares  $R = (C, L)$ , em quantidade suficiente, e um conjunto  $P = \{(c_1, l_1), (c_2, l_2), \dots, (c_n, l_n)\}$  de itens retangulares menores, com as respectivas demandas  $d_i, i = 1, \dots, n$ . Um item  $p_i = (c_i, l_i)$ , inserido no canto esquerdo inferior de  $R$ , determina dois tipos de faixas geradas com a execução de cortes guilhotinas sobre este objeto, conforme ilustrado na figura 2. A execução do corte horizontal a partir do item  $p_i$ , dá origem a uma faixa guilhotina horizontal  $F_H = (C, l_i)$  e um subretângulo  $R_F = (C, L - l_i)$  a ser utilizado, possivelmente, na definição de outras faixas. Já com o corte efetuado na vertical, produz-se uma faixa guilhotina vertical  $F_V = (c_i, L)$  e outro subretângulo  $R_F = (C - c_i, L)$  a ser explorado na configuração das demais faixas do padrão de corte. Estas faixas guilhotinas contidas no padrão de corte, assim

como os itens que a compõem, devem ser sempre posicionadas no canto inferior esquerdo. Diante do exposto, um padrão de corte é distinguido pelo conjunto de faixas guilhotinas em sua configuração e pelo modo como estas faixas são arranjadas.

Na representação dos padrões de corte é utilizada uma sequência de caracteres alfanuméricos, ou seja, as faixas guilhotinas são codificadas como *strings* de caracteres. As faixas guilhotinas, horizontal e vertical, são iniciadas pelos caracteres H e V, nessa ordem, e os itens são representados pela letra P. Por exemplo, na figura 3, a expressão H2P4 indica que a faixa guilhotina horizontal iniciou com três itens  $p_4$ . Da mesma forma, a faixa guilhotina vertical iniciada com  $p_2$  é indicada pela expressão VIP2.

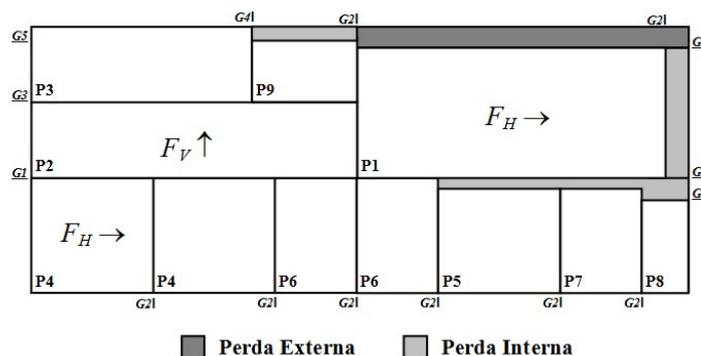


FIGURA 3 – Padrão de corte bidimensional guilhotinado.

Na ocorrência de outros itens na faixa guilhotina, diferentes dos itens envolvidos na sua geração, estes são simbolizadas por M ou A e, estes caracteres distinguem a maneira pela qual o novo item foi escolhido para compor esta faixa. Estes dois últimos caracteres assinalam o tipo de melhoria que foi realizada na faixa guilhotina e a melhoria representada por A é uma extensão do GRASP-2D. Estas melhorias são ilustradas ainda na figura 3, ganhando evidência juntamente com expressões utilizadas para originar as faixas guilhotinas encontradas nesta composição, onde este padrão de corte em cinco estágios pode ser descrito pela *string* de caracteres H2P4M2P6MP5MP7MP8 V1P2A1P3A1P9 H1P1.

Outra questão importante é diferenciar os tipos de perda considerados no padrão de corte (WANG, 1983). O padrão de corte apresenta perda externa associada aos subretângulos  $R_F = (C_F, L_F)$ , com  $C_F \leq C$  e  $L_F \leq L$ , gerados após os cortes que dão origem as faixas guilhotinas  $F_H$  e  $F_V$ , quando estes não podem ser utilizados na construção de novas faixas guilhotinas. Caso existam áreas não aproveitadas em uma faixa guilhotina que compõe o padrão, estas são denominadas perda interna. Portanto, a totalização das medidas de superfície improdutivas no objeto  $R$ , distinguidas em perda interna e perda externa, determinam a perda no padrão de corte. Com o objetivo de elucidar os tipos de perda, a figura 3 coloca em destaque os dois tipos de perda produzidos no padrão de corte.

A seguir, são apresentados os procedimentos característicos da metodologia GRASP, fase de construção e fase de melhoria, desenvolvidos para otimizar a geração de padrões de corte para o PCBGR com valor de utilidade dos itens pertinente as suas respectivas áreas.

#### 4.1 Fase de construção do algoritmo GRASP-2D<sub>A</sub>

Nesta fase, a construção de faixas guilhotinas viáveis é iniciada de forma iterativa, utilizando um item  $p_i$  de cada vez, retirada aleatoriamente da LRC. Sendo a LRC formada pelos itens com demanda ainda não atendida, onde são escolhidos em um processo com três elementos fundamentais: uma lista  $C'$  constituída dos itens aspirantes a originar uma faixa, uma função gulosa  $v: P \rightarrow \mathbb{R}$ , que associa a cada item  $p_i \in P$  um único valor igual à medida de sua área e um critério guloso  $\beta = \max \{v(p_i); p_i \in C'\}$  para determinar o item pertencente a  $C'$  de maior área. Sabendo que  $LRC = \{p_i \in C' / \alpha \cdot \beta \leq v(p_i) \leq \beta\}$  apresenta os itens mais interessantes de  $C'$ , segundo valor de utilidade, e sua cardinalidade é diretamente influenciada pelo parâmetro

adotado  $\alpha \in [0,1]$ , uma seleção de forma aleatória determina o item responsável pela definição das dimensões das novas faixas guilhotinas.

O item  $p_k$  que vir a ser escolhido aleatoriamente na LRC dá início a construção de duas faixas guilhotinas viáveis, através de um corte guilhotina horizontal e outro vertical. Para estas faixas  $F_H$  e  $F_V$ , não só são calculados os referentes  $\eta_k$  números de itens  $p_k$  inseridos e o valor de perda interna associada a cada uma destas, como também são determinados subretângulos  $RF_H = (C_F - \eta_k \cdot c_k, l_k)$  e  $RF_V = (c_k, L_F - \eta_k \cdot l_k)$ , relativos às respectivas perdas internas. Caso o item sorteado não produza uma faixa guilhotina viável no padrão de corte, este item é imediatamente retirado de  $C'$  e as informações relacionadas ao problema são atualizadas para a próxima escolha na LRC.

Se as faixas  $F_H$  e  $F_V$ , construídas de forma sequencial após extração de um item  $p_k$  na LCR, não apresentam perda interna, estas podem ser repetidas no padrão de corte, em quantidades que não extrapolem as dimensões da placa, nem violem a demanda  $d_k$ , de acordo com ajuste feito pelo parâmetro  $\varphi$ . Este parâmetro não foi considerado no GRASP-2D e determina a frequência de iterações que admitem a repetição destas faixas no padrão de corte presente.

#### 4.2 Fase de melhoria do algoritmo GRASP-2D<sub>A</sub>

No procedimento de melhoria, faixas guilhotinas inicialmente construídas, são tratadas com o objetivo de maximizar o valor de utilidade total em ambas, obtendo um melhor aproveitamento dos respectivos subretângulos  $RF_H = (C_H, L_H)$  e  $RF_V = (C_V, L_V)$ . Com a finalidade de evitar movimentos dispensáveis que resultariam em faixas guilhotinas impraticáveis nas dimensões do objeto ou provocariam proibições para execução do corte guilhotinado, o processo de melhoria das faixas é intensificado com a criação dos conjuntos  $B_H$  e  $B_V$ . Os itens pertencentes à  $B_H$ , assim como os incluídos em  $B_V$ , não apresentam dimensões que ultrapassem as faixas  $F_H$  e  $F_V$ , nessa ordem, mantendo assim a viabilidade das faixas quando selecionados para executar melhorias.

Com o propósito de acelerar as ações na fase de melhoria, o conjunto  $B_H$  é iniciado com os itens de  $C'$  que podem ser incluídos no presente subretângulo  $RF_H$ . Considerando o movimento de melhoria simbolizado com o caractere M, o item considerado mais interessante é aquele que apresenta maior largura entre os itens pertencentes a  $B_H$ . Se outros itens pertencentes a este conjunto possuem medidas de largura iguais, uma escolha aleatória define o novo item a compor a faixa guilhotina. Para o movimento de melhoria indicado pelo caractere A, é escolhido o item que possui maior comprimento em  $B_H$  e constatando a ocorrência de mais itens com a mesma dimensão, a seleção também deve ser aleatória.

Em cada uma das estratégias, utilizadas separadamente na melhoria da faixa  $F_H$ , calcula-se o número  $\eta_j$  de itens  $p_j$  a serem incluídos nesta faixa. Decerto que este número de itens está diretamente relacionado às dimensões da perda interna e à sua demanda  $d_j$ , cada inclusão desta maior quantidade possível implica na atualização do conjunto  $B_H$  e do subretângulo  $RF_H$ . Sendo  $RF_H = (C_H - \eta_j \cdot c_j, L_H)$  o subretângulo após utilização de  $p_j$  na melhoria representada por M, ou  $RF_H = (C_H, L_H - \eta_j \cdot l_j)$  depois da melhoria indicada por A. Com o intuito de gerar um melhor aproveitamento da faixa  $F_H$ , este processo é repetido até não existir mais itens em  $B_H$ .

Assim como  $B_H$ , o conjunto  $B_V$  é constituído pelos itens de  $C'$  que, ao serem incluídos no atual subretângulo  $RF_V$ , não impossibilitariam a efetivação da faixa  $F_V$ . As melhorias associadas ao uso dos caracteres M ou A, na expressão utilizada para a faixa  $F_V$ , aplicam-se quando o item escolhido em  $B_V$  apresentar o maior comprimento ou a maior largura, nessa ordem. O número  $\eta_m$  de itens  $p_m$ , a serem inseridos na faixa  $F_V$ , é determinado de acordo com as dimensões da perda interna e a sua demanda  $d_m$ . As melhorias indicadas por M ou A produzem os subretângulos  $RF_V = (C_V, L_V - \eta_m \cdot l_m)$  ou  $RF_V = (C_V - \eta_m \cdot c_m, L_V)$ , respectivamente. O conjunto  $B_V$ , assim como o  $B_H$ , também é atualizado, até não existir mais item que possa melhorar a faixa.

Ao final desta fase, espera-se ter produzido quatro configurações distintas de faixas guilhotinas e, então, aproveitar a faixa que apresenta o menor valor de perda interna na composição do padrão de corte corrente. Considerando que possa acontecer algum empate nos

valores de perda interna entre as duas faixas  $F_H$  e as duas faixas  $F_V$ , a escolha da faixa a configurar no padrão de corte é aleatória.

### 4.3 Pseudocódigo do algoritmo GRASP-2D<sub>A</sub>

Inicialmente, são relacionados os dados de entrada, de saída e as variáveis ainda não explicitadas neste trabalho, para o melhor entendimento do algoritmo a ser descrito posteriormente.

- *maxiter* : número máximo de iterações;
- *pitens* : valor percentual do número de itens utilizados na geração dos padrões;
- *padcorteS\** : melhor padrão de corte;
- *valorS\** : valor de utilidade total do melhor padrão de corte;
- *padcorteS* : padrão de corte corrente;
- *valorS* : valor de utilidade total do padrão de corte corrente;
- *perdaintF<sub>H/V</sub>* : valor da perda interna em uma faixa guilhotina horizontal ou vertical;
- *F<sub>melhor</sub>* : faixa guilhotina incluída em um padrão de corte;
- *perdaintF<sub>melhor</sub>* : valor da perda interna em uma faixa guilhotina.

O pseudocódigo do algoritmo GRASP-2D<sub>A</sub> é descrito como segue.

**Procedimento** GRASP-2D<sub>A</sub> ( $p_i, d_i, v(p_i), R, \alpha, \varphi, maxiter$ )

1.  $padcorteS^* = \emptyset$  e  $valorS^* = 0$ ;
2. Para (Iter = 1, ..., *maxiter*) faça
3.  $padcorteS = \emptyset, valorS = 0$  e defina  $C = (p_1, \dots, p_n)$ ;
4. Enquanto ( $|C| \leq (pitens.n)$ ) faça
5.  $padcorte = \emptyset, R_F = R$  e crie  $C' = (p_1, \dots, p_t), t \leq n$ ;
6. Enquanto ( $|C'| \neq 0$ ) faça
7.  $\beta = \max \{v(p_i); p_i \in C'\}$ ;
8.  $LRC = \{p_i \in C' \mid \alpha.\beta \leq v(p_i) \leq \beta\}$ ;
9. Escolha, aleatoriamente, um item de *LRC* que mantenha a viabilidade;
10. Construa  $F_H$  e  $F_V$ ;
11. Determine  $RF_H, perdaintF_H, RF_V$  e  $perdaintF_V$ ;
12. Se ( $MOD(maxiter, \varphi) = 0$ ) faça
13. Se ( $perdaintF_{H/V} = 0$ ) faça
14. Repita  $F_{H/V}$ ;
15. Fim-se;
16. Fim-se;
17. Crie  $B_H$  e  $B_V$ ;
18. Se ( $|B_H| \neq 0$  e  $|B_V| \neq 0$ ) faça
19. Selecione itens de  $B_H$  e itens de  $B_V$ , para usar "M" e "A";
20. Melhore  $F_H$  e  $F_V$ ;
21. Atualize  $B_H, RF_H, perdaintF_H, B_V, RF_V$  e  $perdaintF_V$ ;
22. Fim-se;
23. Determine  $perdaintF_{melhor} = \min \{perdaintF_H, perdaintF_V\}$ ;
24. Inclua  $F_{melhor}$ , associada à  $perdaintF_{melhor}$ ;
25.  $padcorte = padcorte \cup \{F_{melhor}\}$ ;
26. Atualize  $C'$  e  $R_F = R_F - F_{melhor}$ ;
27. Fim-enquanto;
28. Calcular  $valorS = \sum v_i.x_i$ ;
29. Se  $valorS > valorS^*$  faça
30. Atualize  $padcorteS^* = padcorteS$  e  $valorS^* = valorS$ ;
31. Atualize  $C$ ;
32. Fim-se;
33. Fim-enquanto;
34. Fim-para;
35. Retorna ( $padcorteS^*, valorS^*$ );

**Fim** GRASP-2D<sub>A</sub>

## 5. Algoritmo GRASP-2D<sub>V</sub>

O algoritmo GRASP-2D<sub>V</sub> utiliza praticamente os mesmos alicerces empregados na arquitetura do GRASP-2D<sub>A</sub>, diferenciando-se apenas na maneira como é constituída a LRC e as modificações necessárias nas linhas 7 e 8 do pseudocódigo são descritas a seguir.

É importante ressaltar que as instâncias utilizadas nos testes feitos com o algoritmo GRASP-2D<sub>V</sub> apresentam valor de utilidade  $v_i$ , associado a cada item  $p_i$ , não necessariamente igual à medida de área destes itens, ou seja, existe pelo menos um  $p_i$  tal que  $v_i \neq c_i \cdot l_i$ .

A mudança na concepção da LRC se deve porque a escolha aleatória do item que define a criação das faixas  $F_H$  e  $F_V$  deve considerar, simultaneamente, as grandezas medida de área e valor de utilidade. Sendo assim, o processo de formação da LRC se estabelece com mais três elementos. No início são utilizados elementos que possuem as mesmas características e propósitos dos elementos já utilizados no GRASP-2D<sub>A</sub>, isto é, a lista  $C'$  que continua apresentando os itens com demanda ainda não atendida, uma função  $u: P \rightarrow \mathbb{R}$  calcula a área de cada item e um critério guloso  $\sigma = \max \{u(p_i); p_i \in C'\}$  retorna o item de  $C'$  com maior área, elementos estes que possuem as mesmas características e propósitos dos elementos já utilizados no GRASP-2D<sub>A</sub>. A seguir, é preenchida uma lista  $V = \{p_i \in C' / \alpha \cdot \sigma \leq u(p_i) \leq \sigma\}$  para ser utilizada juntamente com a função gulosa  $v: P \rightarrow \mathbb{R}$ , onde  $v(p_i)$  é o valor de utilidade de cada item  $i$ , na definição do critério guloso  $\beta = \max \{v(p_i); p_i \in V\}$ . Observe que na escolha dos itens para compor  $V$ , apenas a grandeza área é relevante e o parâmetro  $\alpha$  se encarrega de sua cardinalidade. Ao término desse processo, um item é escolhido aleatoriamente na  $LRC = \{p_i \in V / \alpha \cdot \beta \leq v(p_i) \leq \beta\}$ , constituída dos itens mais interessantes de  $V$ , para dar origem as novas faixas guilhotinas.

## 6. Resultados Computacionais

Para analisar o desempenho dos algoritmos GRASP-2D<sub>A</sub> e GRASP-2D<sub>V</sub> foram realizados testes para um conjunto de 30 instâncias extensamente utilizadas na literatura, com soluções ótimas conhecidas, e disponibilizadas por Morabito e Pureza (2010) em ESICUP<sup>1</sup>. Estas instâncias estão divididas em dois grupos que consideram a distinção dos valores de utilidade. As 16 instâncias do primeiro grupo, com valor de utilidade dos itens igual a sua área, são: WANG1, WANG2, WANG3, OF1, OF2 e CU1-CU11. O segundo grupo com 14 instâncias e item apresentando valor de utilidade diferente de sua área, é constituído das seguintes: ChW1, ChW2, ChW3 e CW1-CW11. Os algoritmos foram implementados em linguagem C/C++ e os testes foram realizados em um computador com processador Intel(R) Core(TM)i5 1.7 GHz, com 4GB de memória RAM e sistema operacional Windows 8 de 64 bits.

Sabendo que os valores de função objetivo informam o valor de utilidade total do padrão de corte, as tabelas 1 e 2 apresentam os resultados obtidos em 10 execuções com os algoritmos GRASP-2D<sub>A</sub> e GRASP-2D<sub>V</sub>, respectivamente. Na variação do parâmetro alfa  $\alpha$ , adotaram-se os valores 0.1, 0.25, 0.5, 0.75 e 1.0. O critério de parada foi estabelecido em 10000 iterações, com as primeiras 5000 utilizando até 50% dos itens na geração dos padrões de corte e aplicando o limite de 10% nas seguintes. O parâmetro  $\phi = 5$  ficou fixado em todas as rodadas. Os tempos computacionais para a melhor solução ( $t$ ), execução total ( $tt$ ) e tempo médio de execução ( $ttm$ ) são apresentados em segundos.

Os resultados encontrados nas tabelas 1 e 2 estão sintetizados na tabela 3. Nesta tabela, os melhores resultados são confrontados com os apresentados pelos algoritmos propostos por Fayard *et al.* (1998), Alvarez-Valdés *et al.* (2002), Hifi (2004) e Morabito e Pureza (2010). Sobre estes trabalhos, Fayard *et al.* (1998) utiliza programação dinâmica com a estratégia para dois estágios de Gilmore e Gomory (1965), Alvarez-Valdés *et al.* (2002) desenvolvem dois algoritmos, um sofisticado Busca Tabu e outro GRASP, Morabito e Pureza (2010) apresentam um procedimento heurístico que emprega programação dinâmica e busca em grafo E/OU e Hifi (2004) propõe um algoritmo híbrido que utiliza *hill-climbing* e programação dinâmica. Estes algoritmos são indicados por FHZ98, APT02, H04 e DP\_AOG10, utilizando uma notação que ressalta o ano das respectivas publicações.

<sup>1</sup> <http://paginas.fe.up.pt/~esicup/tiki-index.php>



É importante observar, ainda na tabela 3, que os resultados expostos destacam as soluções ótimas denotadas com \*, a diferença relativa entre o ótimo e o valor de função objetivo do melhor padrão de corte obtido com os respectivos algoritmos, o parâmetro alfa e os respectivos tempos envolvidos nessa execução. Em algumas instâncias, as soluções não disponibilizadas nos artigos são indicadas com -.

TABELA 1 – Resultados do primeiro grupo de instâncias com o GRASP-2D<sub>A</sub>.

Alfa	Instância	WANG1	WANG2	WANG3	OF1	OF2	CU1	CU2	CU3	CU4	CU5	CU6	CU7	CU8	CU9	CU10	CU11
		n	20	20	20	10	10	25	35	45	45	50	45	25	35	25	40
0,1	G-2DA	2277	2694	2721	2737	2690	12330	26100	16679	99366	173010	158572	247150	432714	657055	771102	920436
	t	0	0,828	0,016	0,156	0,062	0,125	0,437	6,376	17,563	1,797	0,172	0,984	0,328	0,063	11,907	24,657
	tt	15,61	7,797	8,5	6,094	5,047	15,892	20,501	39,69	23,173	27,955	24,783	11,25	15,876	16,486	25,298	26,986
	G-2Dam	2277,00	2694,00	2721,00	2737,00	2690,00	12330,00	26100,00	16641,70	99050,30	173010,00	158572,00	246842,10	432396,40	657055,00	769257,00	916126,30
	ttm	15,690	7,793	8,729	6,199	5,174	17,529	20,289	36,532	22,739	30,324	26,150	12,484	15,984	17,026	25,012	28,730
0,25	G-2DA	2277	2544	2721	2737	2690	12330	26100	16583	98859	173010	158572	247150	431126	657055	772982	917467
	t	0	0	0,016	0,016	0	0,235	0,172	0,265	3,859	1,078	1,218	0,422	0,093	0,188	7,063	24,502
	tt	12,923	7,485	7,625	5,969	5,344	16,189	18,142	31,892	21,751	28,704	23,548	10,86	15,938	17,954	25,673	28,142
	G-2Dam	2277,00	2544,00	2721,00	2737,00	2690,00	12330,00	26100,00	16605,00	99080,90	172996,40	158572,00	246571,40	431126,00	657055,00	769921,80	915424,40
	ttm	15,695	7,388	7,880	6,175	5,163	17,686	18,682	33,507	22,590	28,619	23,158	12,091	16,271	18,156	26,598	31,633
0,5	G-2DA	2277	2544	2623	2737	2690	12330	25396	16583	98452	173004	158572	246075	431126	657055	772118	916194
	t	0	0	0	0,016	0,25	0,016	0,015	0,031	3,235	19,767	1,078	3,704	0,89	0	0,766	12,845
	tt	8,704	7,047	7,376	6,547	4,938	12,860	15,641	26,673	17,783	26,673	20,423	12,173	14,126	16,423	25,064	32,627
	G-2Dam	2277,00	2544,00	2623,00	2737,00	2690,00	12330,00	25423,40	16592,50	98804,20	171953,60	158414,50	245853,60	430822,80	657055,00	770906,80	915735,40
	ttm	10,060	7,332	7,458	6,491	5,041	15,432	17,310	27,487	19,395	26,047	20,004	12,624	14,216	17,256	24,737	32,758
0,75	G-2DA	2277	2509	2549	2670	2515	12330	25396	16583	98015	171278	158047	245570	428372	657055	772118	916652
	t	0,015	0,016	0	0	0,031	0,031	0,047	0,14	0,25	4,922	0	0,578	0,031	0	3,813	11,563
	tt	8,641	6,923	7,079	5,484	5,125	16,204	13,751	24,048	17,345	25,314	18,501	11,891	13,844	16,266	23,533	29,454
	G-2Dam	2277,00	2509,00	2549,00	2670,00	2515,00	12330,00	25399,60	16583,00	98015,00	170356,90	158047,00	245120,10	428372,00	657055,00	769447,80	914996,20
	ttm	9,529	6,957	7,247	5,628	5,246	12,610	14,063	24,936	17,859	25,190	19,881	10,857	13,702	16,653	23,919	30,013
1	G-2DA	2093	2238	2470	2290	2346	11805	24931	16254	97482	168071	153351	240812	404176	657055	700481	859944
	t	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	tt	9,516	7,813	7,531	5,313	4,735	10,298	12,860	19,173	17,36	22,563	17,235	10,641	12,063	14,142	24,392	26,876
	G-2Dam	2093,00	2238,00	2470,00	2290,00	2346,00	11805,00	24931,00	16254,00	97482,00	168071,00	153351,00	240812,00	404176,00	657055,00	700481,00	859944,00
	ttm	9,636	8,235	7,615	5,444	4,814	10,551	14,984	23,070	17,445	23,947	17,287	11,791	12,266	15,187	25,187	27,445

TABELA 2 – Resultados do segundo grupo de instâncias com o GRASP-2D<sub>V</sub>.

Alfa	Instância	ChW1	ChW2	ChW3	CW1	CW2	CW3	CW4	CW5	CW6	CW7	CW8	CW9	CW10	CW11
		n	7	10	20	25	35	40	39	35	55	45	60	50	60
0,1	G-2DV	244	2892	1860	6402	5354	5689	6170	11659	12635	9898	4464	10748	6212	6084
	t	0	2,876	0,468	0,218	3,703	22,829	21,783	1,75	33,502	21,689	9,86	16,297	25,829	4,687
	tt	4,672	5,798	19,344	19,782	16,829	28,798	26,502	16,548	51,331	36,362	52,66	45,924	51,596	41,955
	G-2DVm	244,00	2822,00	1860,00	6402,00	5245,30	5606,50	6157,40	11569,40	11632,20	9658,40	4268,70	10169,70	6003,40	5876,00
	ttm	4,752	5,888	19,098	17,101	14,057	28,073	26,267	17,426	48,685	35,094	53,440	44,363	51,782	41,271
0,25	G-2DV	244	2826	1720	6402	5279	5672	4817	10732	8277	8260	4229	8504	6084	6084
	t	0	0,031	4,001	4,876	7,173	1,438	13,438	11,985	9,11	0,547	37,799	3,829	45,518	0,421
	tt	4,953	6,188	19,861	16,298	12,392	28,549	22,923	15,298	41,58	33,892	50,8	39,549	51,081	41,767
	G-2DVm	244,00	2823,60	1682,00	5978,90	5235,10	5622,20	4732,60	8954,30	7825,70	8126,60	4162,00	8231,30	5997,60	5865,90
	ttm	5,078	6,322	20,371	16,390	14,749	27,330	24,123	16,609	41,740	34,733	50,751	41,899	52,495	43,655
0,5	G-2DV	244	2749	1540	5373	4950	5466	4372	7851	7771	6765	3929	5890	4749	4574
	t	0	0,016	0,515	3,5	6,406	9,641	1,328	3,219	12,392	0,594	9,907	18,704	16,486	15,954
	tt	5,407	6,516	22,376	13,657	11,907	23,986	15,079	14,204	35,83	30,173	48,268	39,393	45,362	37,408
	G-2DVm	244,00	2749,00	1540,00	5247,30	4834,10	5292,10	4298,10	7267,00	7286,10	6160,80	3203,80	5845,00	4478,70	4505,40
	ttm	5,628	6,594	23,001	15,640	13,027	27,023	18,857	14,874	37,058	30,152	47,588	38,141	45,651	37,299
0,75	G-2DV	244	2514	1340	4343	3659	4795	3635	5823	6430	5336	3026	4490	3813	3495
	t	0	0	0,781	0	0,219	1,359	0,015	0,141	29,939	0,016	1,312	0,985	0,016	0,062
	tt	6,813	6,766	20,908	11,532	13,251	26,439	20,345	16,673	36,486	29,095	45,143	35,127	41,221	39,596
	G-2DVm	244,00	2514,00	1322,00	4343,00	3659,00	4760,30	3635,00	5823,00	5867,10	5336,00	2994,40	4438,80	3813,00	3495,00
	ttm	6,805	6,813	21,057	13,230	13,215	27,259	19,932	17,445	37,108	28,873	45,046	36,527	41,479	40,709
1	G-2DV	244	2460	960	3152	2758	3495	2652	3476	3325	3847	1748	3548	3313	3378
	t	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	tt	6,172	8,392	17,516	11,595	11,485	23,033	16,391	13,329	36,392	23,236	42,486	35,643	37,08	33,548
	G-2DVm	244,00	2460,00	960,00	3152,00	2758,00	3495,00	2652,00	3476,00	3325,00	3847,00	1748,00	3548,00	3313,00	3378,00
	ttm	6,510	8,424	17,820	11,769	12,227	23,690	16,954	14,713	39,507	27,308	49,034	40,218	41,198	35,989

TABELA 3 – Resultados dos algoritmos GRASP-2D<sub>A</sub> e GRASP-2D<sub>V</sub>.

Instância	n	FHZ98	APT02	H04	DP_AOG10	GRASP-2D <sub>A/V</sub>	gap	Alfa	t	tt
WANG1	20	-	-	2277*	2277*	2277*	0	0,5	0,000	8,704
WANG2	20	-	-	2694*	2694*	2694*	0	0,1	0,828	7,797
WANG3	20	2721*	2721*	2721*	2721*	2721*	0	0,25	0,016	7,625
OF1	10	2713	2737*	2737*	2737*	2737*	0	0,25	0,016	5,969
OF2	10	2586	2690*	2690*	2690*	2690*	0	0,25	0,000	5,344
CU1	2,5	12312	12330*	12330*	12330*	12330*	0	0,75	0,031	16,204
CU2	3,5	25806	26100*	26100*	26100*	26100*	0	0,25	0,172	18,142
CU3	4,5	16608	16679	16723*	16723*	16679	0,003	0,1	6,376	39,690
CU4	4,5	98190	99366	99495*	99495*	99366	0,001	0,1	17,563	23,173
CU5	5,0	171651	173364*	173364*	173364*	173010	0,002	0,25	1,078	28,704
CU6	4,5	158572*	158572*	158572*	158572*	158572*	0	0,1	0,172	24,783
CU7	2,5	246860	247150*	247150*	247150*	247150*	0	0,25	0,422	10,860
CU8	3,5	432198	432714	433331*	433331*	432714	0,001	0,1	0,328	15,876
CU9	2,5	657055*	657055*	657055*	657055*	657055*	0	1	0	14,142
CU10	4,0	764696	773485	773772*	773772*	772982	0,001	0,25	7,063	25,673
CU11	5,0	913387	922161	924696*	924696*	920436	0,005	0,1	24,657	26,986
ChW1	7	-	-	244*	244*	244*	0	0,1	0	4,672
ChW2	10	2731	2892*	2892*	2892*	2892*	0	0,1	2,876	5,798
ChW3	20	1740	1860*	1860*	1860*	1860*	0	0,1	0,468	19,344
CW1	2,5	6402*	6402*	6402*	6402*	6402*	0	0,1	0,218	19,782
CW2	3,5	5354*	5354*	5354*	5354*	5354*	0	0,1	3,703	16,829
CW3	4,0	5148	5689*	5689*	5689*	5689*	0	0,1	22,829	28,798
CW4	3,9	6168	6170	6175*	6175*	6170	0,001	0,1	21,783	26,502
CW5	3,5	11550	11644	11659*	11659*	11659*	0	0,1	1,750	16,548
CW6	5,5	12403	12923*	12923*	12923*	12635	0,022	0,1	33,502	51,331
CW7	4,5	9484	9898*	9898*	9898*	9898*	0	0,1	21,689	36,362
CW8	6,0	4505	4605*	4605*	4605*	4464	0,031	0,1	9,860	52,660
CW9	5,0	10748*	10748*	10748*	10748*	10748*	0	0,1	16,297	45,924
CW10	6,0	6116	6515*	6515*	6515*	6212	0,046	0,1	25,829	51,596
CW11	6,0	6084	6321*	6321*	6321*	6084	0,037	0,25	0,421	41,767

Analisando os resultados mostrados na tabela 3, verifica-se que nas dezesseis instâncias do primeiro grupo, o algoritmo GRASP-2D<sub>A</sub> encontrou dez soluções ótimas e as demais apresentaram gap inferior a 0,005. Nestas seis instâncias remanescentes, três apresentaram mesmo valor de função objetivo do algoritmo APT02. Observa-se, também, que em nenhum caso o algoritmo FHZ98 apresentou resultado superior do que os obtidos com o GRASP-2D<sub>A</sub>. Os resultados para o segundo grupo, constituído de quatorze instâncias, mostram que o GRASP-2D<sub>V</sub> conseguiu nove soluções ótimas, com gap igual a 0,046 na pior ocorrência. Ainda que os tempos computacionais fornecidos pelos algoritmos sejam confrontáveis, mas não discutidos nesse trabalho, o ótimo obtido na instância CW5 indica competitividade do algoritmo GRASP-2D<sub>V</sub> quando comparado às soluções apresentadas com APT02 e FHZ98. Das cinco instâncias sem ótimo na resposta para este grupo, duas exibiram mesmo valor de função objetivo de APT02 e FHZ98, e outras duas superaram apenas FHZ98. Nas instâncias com gap indesejável, pressupõe que uma aferição pormenorizada dos parâmetros, quando utilizados em instâncias com grande quantidade de itens distintos, se faz necessário para obter resultados satisfatórios com os algoritmos propostos.

Na figura 4, alguns dos padrões de corte obtidos com a execução dos algoritmos GRASP-2D<sub>A</sub> e GRASP-2D<sub>V</sub> estão ilustrados, com as consideradas proporções. As respectivas representações em *strings* de caracteres estão discriminadas abaixo.

- (a) V2P8 V1P2 H1P8A1P7A1P1 V1P6 H2P5 H1P10
- (b) V1P2A1P5 V2P4M1P6 V2P1M1P3
- (c) V2P25A1P33 H2P34 V2P31M1P34 H1P4 H1P4 V1P17
- (d) H3P17 H3P17 V2P25 H1P4A1P25 V1P17 V1P17

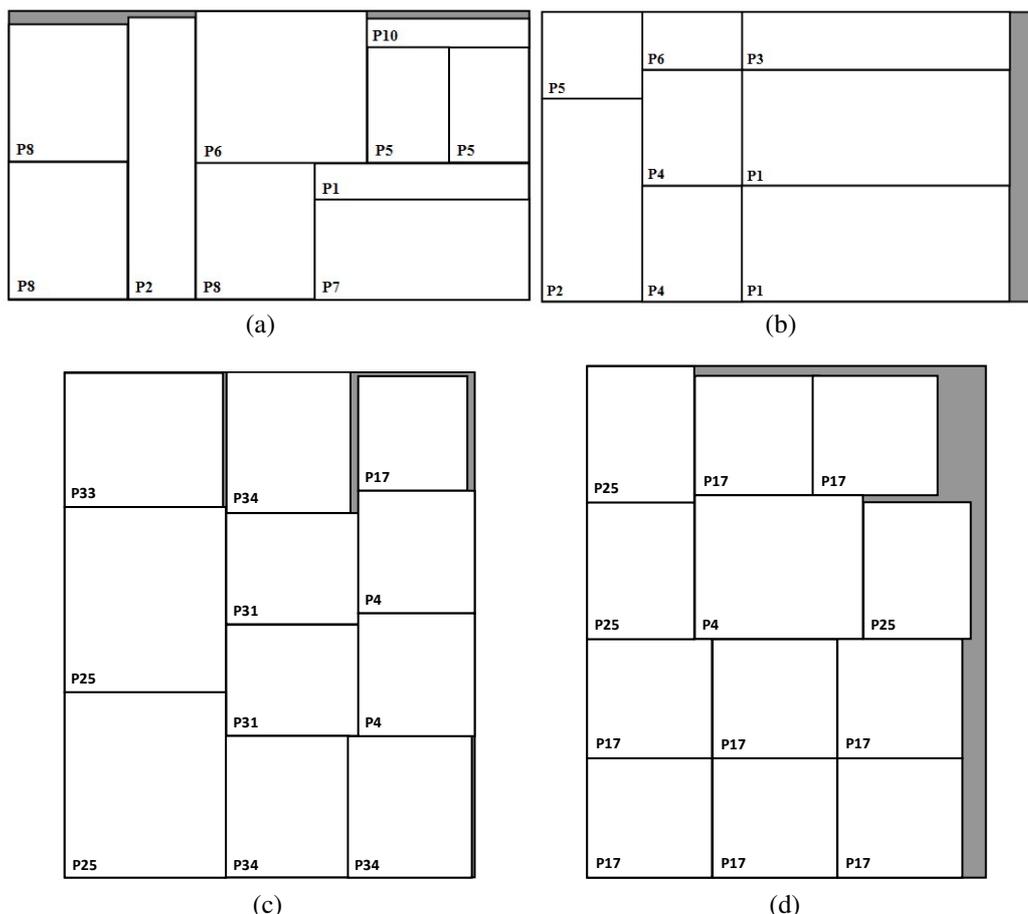


FIGURA 4 – Padrões de Corte OF1(a), ChW1(b), CU4(c) e CW5(d).

## 7. Conclusões

Neste trabalho foram apresentados os algoritmos, baseados na metodologia GRASP e concebidos a partir de ampliações no algoritmo GRASP-2D, nas versões GRASP-2D<sub>A</sub> e GRASP-2D<sub>V</sub>, para geração de padrões de corte bidimensionais guilhotinados restrito.

Com estratégias que diferem de acordo com o valor de utilidade dos itens, os algoritmos foram analisados separadamente em dois grupos de instâncias clássicas e os seus resultados foram comparados aos valores das soluções de algoritmos competitivos e bastante conhecidos na literatura. Os resultados foram promissores, conseguindo uma quantidade significativa de dezenove soluções ótimas, nas trinta instâncias analisadas, e as demais apresentando um gap pequeno, com tempos computacionais de execução procedentes, em média inferiores a um minuto.

É importante ressaltar que a definição dos valores, percentual do número de itens utilizados na geração dos padrões e para o parâmetro  $\phi$ , foram importantes, mas a variação dos valores atribuídos ao parâmetro  $\alpha$  foi decisiva, influenciando o resultado final em algumas instâncias. As instâncias do primeiro grupo foram menos sensíveis a variação do  $\alpha$  no GRASP-2D<sub>A</sub> do que as instâncias do segundo grupo, com  $\alpha$  igual a 0,1 encontrando melhores soluções no GRASP-2D<sub>V</sub>. Assim, é possível concluir que o valor do parâmetro  $\alpha$  tem influencia direta na qualidade da solução e uma aferição precisa deste parâmetro pode produzir alterações significativas nos resultados.

Como perspectivas futuras, a combinação de técnicas exatas e outras metodologias heurísticas devem ser consideradas nas novas ampliações do GRASP-2D para resolução dos Problemas de Corte e Empacotamento.

### Referências Bibliográficas

- Alvarez-Valdés, R., Parajón, A., and Tamarit, J.** (2002). A tabu search algorithm for large-scale guillotine (un)constrained two-dimensional cutting problems. *Computers and Operations Research*, v.29, p.925-947.
- Amaral, A.R.S. and Wright, M.** (2001), Efficient algorithm for the constrained two-dimensional cutting stock problem, *International Transactions in Operational Research*, v.8, p.3-13.
- Dyckhoff, H.** (1990). A Typology of Cutting and Packing Problems. *European Journal of Operational Research*, v.44, p.145-159.
- ESICUP.** Euro Special Interest Group on Cutting and Packing. Disponível em: <<http://paginas.fe.up.pt/~esicup/>> (último acesso 14/05/2014).
- Fayard, D., Hifi, M., and Zissimopoulos, V.** (1998). An efficient approach for large-scale two-dimensional guillotine cutting stock problems. *Journal of the Operational Research Society*, v.49, p.1270-1277.
- Feo, T. A. and Resende, M. G. C.** (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operation Research Letters*, n.8, p.67-71.
- Gilmore, P.C. and Gomory, R.E.** (1965). Multistage cutting problems of two and more dimensions, *Operations Research*, v.13, p. 94-119.
- Hifi, M.** (2004), Dynamic programming and hill-climbing techniques for constrained twodimensional cutting stock problems. *Journal of Combinatorial Optimization* 8, 65-84.
- Morabito, R. and Pureza, V.** (2010) A heuristic approach based on dynamic programming and and/or-graph search for the constrained two-dimensional guillotine cutting problem. *Annals of Operation Research*, v.179, p. 297-315.
- Oliveira, J. F. and Ferreira, J. S.** (1990). An improved version of Wang's algorithm for two-dimensional cutting problems. *European Journal of Operational Research*, v.44, p.256-266.
- Silva, A.T.S., Aloise, D.J., Lima Junior, F.C., Oliveira, F.M., Figueiredo, J.N., e Mariano, M.F.S.** (2013), *Proposta de Melhoria no Algoritmo GRASP-2D aplicado ao Problema do Corte Guilhotinado e Restrito*. In: XLVSBPO - Simpósio Brasileiro de Pesquisa Operacional, Natal.
- Velasco, A.S.** (2005), *GRASP para o Problema de Corte Bidimensional Guilhotinado e Restrito*. Dissertação (Mestrado em Engenharia de Produção) – Centro de Ciências e Tecnologia, Laboratório de Engenharia de Produção – Leprod, UENF, Campos dos Goytacazes.
- Velasco, A.S., Paula Junior, G.G. e Vieira Neto, E.** (2008), Um Algoritmo Heurístico Baseado na GRASP para o Problema de Corte Bidimensional Guilhotinado e Restrito. *Revista Gepsos - Gestão da Produção, Operações e Sistemas*, ed.1, p.129-141.
- Wang, P.Y.** (1983), Two algorithms for constrained two-dimensional cutting stock problems. *Operations Research*, v.31, p.573-586.