# ALGORITMOS GRASP HÍBRIDOS PARA O PROBLEMA DE CORTE UNIDIMENSIONAL

**Euclydes Vieira Neto**
Institutos Superiores de Ensino do Censa - ISECENSA
Rua Salvador Correa, 139. Centro - Campos dos Goytacazes - RJ. CEP 28035-310
euclydes@censanet.com.br

**André Soares Velasco**
Instituto Federal Fluminense - IFF
Av. Souza Mota, 350. Parque Fundão - Campos dos Goytacazes - RJ. CEP: 28060-010
asvelasco@iff.edu.br

**Geraldo Galdino de Paula Junior**
Universidade Estadual do Norte Fluminense - UENF
Av. Alberto Lamego, 2000. Parque Califórnia - Campos dos Goytacazes - RJ. CEP: 28013-602
galdino@uenf.br

## RESUMO

Este trabalho apresenta um estudo sobre algoritmos híbridos, a partir da metaheurística GRASP, para o problema de corte unidimensional. Os algoritmos GRASP-1D, ALG HB1 e ALG HB2 são compostos de duas fases, uma de construção e outra de melhoria, com diferentes procedimentos de busca local. O GRASP-1D difere-se dos demais porque é realizada a busca local a cada iteração, isto é, para cada solução inicial. Já os algoritmos ALG HB1 e ALG HB2, registram as melhores soluções iniciais da fase de construção e só diferem na escolha dos esquemas de corte que serão selecionados para o procedimento de melhoria. Os algoritmos foram bem testados computacionalmente através de instâncias práticas retiradas da indústria. Os resultados obtidos são analisados, indicando um desempenho bastante satisfatório do algoritmo ALG HB2.

**PALAVARAS CHAVE. GRASP. Corte unidimensional. Esquema de Corte.**

## ABSTRACT

This work presents a study about hybrid algorithms from GRASP meta-heuristic to the one-dimensional cutting stock problem. The algorithm GRASP-1D, ALG HB1 and ALG HB2 are composed of two phases, one of construction and other of improvement, with different local search procedures. The GRASP-1D differs from the other ones because the local search is carried out every each iteration, that is, for each initial solution. On the other hand, the algorithms ALG HB1 and ALG HB2 register better initial solutions of the construction phase and only differ on the choice of the patterns cut that will be selected to the improvement procedure. The algorithms were computationally well tested through the use of common industrial instances. The obtained results are analyzed, indicating such a satisfactory performance of the algorithm ALG HB2.

**KEYWORDS. GRASP. One-dimensional cutting stock. Pattern cut.**

## 1. Introdução

Em grande parte dos processos de produção nos segmentos industriais de metal-mecânica, cartonagem, construção civil, móveis, entre outros, o corte é uma fase de grande relevância, onde se buscam aperfeiçoamentos que possam melhorar a produtividade e a qualidade dos processos ou dos produtos.

No problema do corte unidimensional, o comprimento é a única dimensão relevante na construção dos esquemas de corte que irão compor uma solução. A figura 1 ilustra um problema de corte unidimensional, onde se tem um número ilimitado de barras padrão em estoque, de comprimento 100, e o objetivo é cortar estas em várias barras menores, com quantidades pré-fixadas de comprimento, que variam de 5 a 46. Estes dois grupos, o de barras em estoque (objetos) e os de barras demanda (itens), são os dados básicos do problema de corte. Uma solução para o problema busca fornecer esquemas de corte, que venham a atender uma demanda de itens e apresentem perda mínima de material durante o processo (DYCKHOFF, 1990).

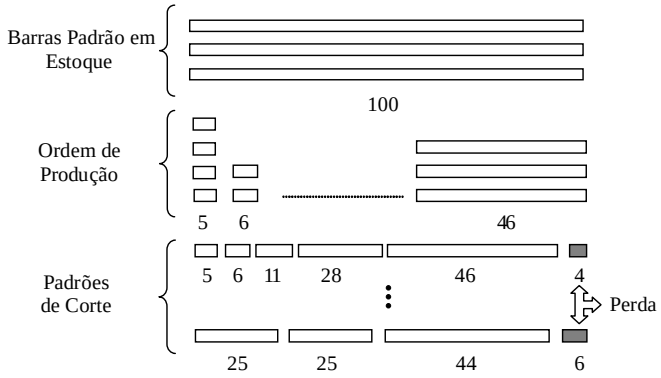


FIGURA 1 - Esquema de corte unidimensional.

A determinação da solução ótima para este problema está relacionada a um processo de otimização combinatória, que acarreta em intratabilidade do ponto de vista computacional. Esta dificuldade se deve ao grande número de esquemas de corte possíveis; além disso, o desejo de enumerá-los, em busca da solução ótima, é inviável do ponto de vista prático. Pertencendo a classe de problemas denominada NP-Difícil, o uso de algoritmos exatos para solução deste problema de corte, necessita normalmente de um tempo computacional considerável, inviabilizando a sua utilização em situações práticas, mesmo para problemas considerados pequenos. Levando em consideração o esforço computacional demandado para o processamento da solução, uma vez que o mercado é muito dinâmico, no que se refere às ofertas de matéria-prima e às alterações de projetos, o uso de heurísticas tem sido a opção preferida.

Considerando a tendência de resolver o problema de corte unidimensional com o uso de heurísticas e algoritmos híbridos, algumas destas abordagens podem ser encontradas em: Gilmore e Gomory (1961); Haessler (1975); Stadtler (1990); Rocha (1997); Cintra (1998); Foerster e Wäscher (2000); Vieira Neto (2004); Yanasse e Limeira (2006); Cui et al. (2008); Yanasse e Cerqueira (2009).

A característica apresentada pela metaheurística GRASP (*Greedy Randomized Adaptive Search Procedures*) na construção de soluções independentes foi a grande motivação do desenvolvimento deste trabalho, que teve como objetivo avançar na pesquisa feita sobre a influência do parâmetro de aleatoriedade no processo de construção das soluções iniciais para o problema do corte unidimensional (VIEIRA NETO, VELASCO e PAULA JUNIOR, 2010).

Considerando a hipótese que entre várias soluções iniciais diferentes para o mesmo problema do corte, nem sempre a melhor solução inicial submetida a um tratamento de melhoria resulta em uma melhor solução final, foi realizada esta pesquisa experimental, executando algoritmos híbridos, designados pelos nomes GRASP-1D, ALG HB1 e ALG HB2, em instâncias obtidas de projetos industriais, para verificar o comportamento da metodologia GRASP na comprovação desta hipótese.

## 2. Formulação do Problema

Ao considerar um estoque de barras padrão de comprimento L (objetos), sem limitação no número de p disponíveis, o problema será cortá-las de maneira que o pedido de $d_i$ barras (itens), com demanda de comprimento $l_i$ (i = 1, 2, ..., m), seja atendido, minimizando o número de barras padrão utilizadas nesse processo de corte. Desta forma, se houver perda em cada barra padrão, a perda total será a menor possível.

Conhecendo os esquemas de corte executáveis, o problema pode ser modelado como um problema de Programação Linear e a sua formulação matemática básica tem a seguinte estrutura:

$$\text{Min } Z = \sum_{j=1}^{n} x_j$$

$$\text{s.a : } \sum_{j=1}^{n} a_{ij}\, x_j = d_i \;,\; i = 1, ..., m$$

$$x_j \geq 0 \text{ e Inteiro}, \; j = 1, ..., n$$

em que:

- $x_j$ é número de vezes que o esquema de corte j é executado;
- $a_{ij}$ é número de vezes que o item i será cortado no esquema de corte j;
- $d_i$ é demanda do item i;
- m é o número de itens;
- n é o número de esquemas de corte viáveis.

Considerando a integridade das variáveis $x_j$ e a dificuldade de apresentar todos os esquemas executáveis com o aumento do número de itens e suas respectivas demandas, estes problemas são reconhecidamente tidos como NP-Difícil (McDIARMID, 1999). Sendo assim, abandonar a utilização de métodos exatos para resolução deste problema em prol das heurísticas, se torna exequível nestas circunstâncias.

## 3. GRASP

A metodologia GRASP foi idealizada como a combinação de uma heurística construtiva e um procedimento de busca local. Um pseudocódigo genérico GRASP é descrito a seguir (FEO e RESENDE, 1995).

**Procedimento** *GRASP* ( )

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

**Fim** *GRASP*

É importante destacar que para construir uma solução, em vez de considerar apenas um candidato disponível para a escolha, ela cria um conjunto de candidatos, dentre os quais apenas um será escolhido aleatoriamente; desta maneira, ela pode construir várias soluções diferentes e aplicar um procedimento de busca local, com o objetivo de melhorar cada uma destas soluções iniciais. O algoritmo GRASP é projetado em duas fases a cada iteração: a fase de construção, seguida da fase de melhoria.

## 4. Algoritmo Grasp-1D

A escolha do algoritmo guloso FFD (*First Fit Decreasing*), para o procedimento de construção das soluções iniciais do algoritmo GRASP-1D (VIEIRA NETO, 2004), deve-se ao fato de este ser um algoritmo de fácil entendimento e implementação, apresentando resultados bastante satisfatórios na otimização de cortes unidimensionais, com perdas médias na ordem de 10% (ROCHA, 1997).

As variáveis envolvidas no algoritmo GRASP-1D e seu pseudocódigo serão apresentadas a seguir. Em relação às variáveis, têm-se:

- m - número de itens;
- L - comprimento da barra padrão do estoque;
- $l_i$ - comprimento do item i;
- $d_i$ - demanda do item i;
- $\alpha$ - parâmetro de aleatoriedade;
- Solução - esquemas de corte;
- Perda - comprimento não utilizado da barra padrão;
- BarrasUsadas - barras padrão utilizadas.

O pseudocódigo GRASP-1D é descrito como segue.

**Procedimento** GRASP-1D ($m, l_1 \ldots l_m, d_1 \ldots d_m, \alpha, L, n_{iterações}$)

1. Se $i_{teração} = n_{iterações}$, ir para o passo 6;
2. Executar ALG_SOLUÇÃO INICIAL, gerando uma solução inicial $S_{inicial}$;
3. Executar ALG_MELHORIA, gerando uma solução melhorada $S_{melhorada}$;
4. Registrar a melhor solução;
5. Voltar ao passo 1;
6. Atribuir $Solução_{final} = Solução_{melhorada}$;
7. Atribuir $Perda_{total} = Perda_{melhorada}$;
8. Atribuir $BarrasUsadas_{total} = BarrasUsadas_{melhor}$;
9. Escrever ($Solução_{final}$, $Perda_{total}$, $BarrasUsadas_{total}$);

**Fim** GRASP-1D

### 4.1 Procedimento para Construir Soluções Iniciais

O Procedimento para Construir Soluções Iniciais, que será tratado por ALG_SOLUÇÃO INICIAL, é executado a cada iteração do GRASP-1D. Vale destacar que as características herdadas pela metodologia GRASP para fase de construção da solução, possibilitam criar soluções diferentes e independentes, o que torna esta fase muito importante e interessante para a solução final de um problema do corte unidimensional.

A seguir, é apresentado o pseudocódigo ALG_SOLUÇÃO INICIAL.

**Procedimento** ALG_SOLUÇÃO INICIAL ($m, l_1 \ldots l_m, d_1 \ldots d_m, \alpha, L$)

1. Criar N dos índices onde N = {1, 2, ..., m};
2. Se N = ∅, ir para o passo 20;
3. Criar conjunto de índices Na, sendo Na = N;
4. Considerar Perda = L;
5. Identificar o item $l_i$ de menor comprimento ($l_{menor}$) para $i \in Na$: $l_{menor} := \min \{l_i : i \in Na\}$ ;
6. Se Na = ∅ (ou Perda < $l_{menor}$), o esquema de corte está finalizado ir para passo 16;
7. Identificar o β que é o item $l_i$ de maior comprimento: $\beta = \max \{l_i : i \in Na\}$;

8. Criar LRC: LRC = $\{i \in Na: l_i \geq \alpha.\beta\}$;
9. Escolher randomicamente um candidato (k) do LRC: k = Randon (LRC);
10. Se Perda < $l_k$ ir para passo 13;
11. Calcular quantas vezes (VezesItem) o item $l_k$ pode ser cortado na barra padrão: $VezesItem_i = \min \{d_k; \lfloor Perda/l_k \rfloor\}$;
12. Calcular o novo valor da perda após cortar o item k: $Perda = Perda - VezesItem_i.l_k$;
13. Retirar o k do conjunto Na: Na = Na - {k};
14. Identificar novamente o item $l_i$ de menor comprimento ($l_{menor}$) para $i \in Na$;
15. Voltar ao passo 6;
16. Verificar quantas vezes o esquema de corte ($Executa_{esquema}$) pode ser executado: $Executa_{esquema} = \min \{d_i/VezesItem_i: VezesItem_i > 0\}$;
17. Atualizar demandas: $d_i = d_i - Executa_{esquema}.VezesItem_i$;
18. Se $d_i = 0$, retirar o item i do conjunto N : N = N - {i};
19. Voltar ao passo 2;
20. Calcular o número total de barras padrão usadas, simbolizada por $BarrasUsadas_{inicial}$: $BarrasUsadas_{inicial} = Soma\ (Executa_{esquema})$;
21. Escrever($Solução_{final}$, $Perda_{Total}$, $BarrasUsadas_{inicial}$);

**Fim** ALG_SOLUÇÃO INICIAL

### 4.2 Procedimento de Melhoria da Solução

O procedimento de melhoria tem como finalidade o tratamento da solução inicial encontrada a cada iteração. Como a GRASP nos proporciona conseguir várias soluções iniciais diferentes para o mesmo problema, é possível que alguma destas soluções, após serem submetidas a este procedimento, alcance um melhor resultado.

Neste procedimento, de cada solução inicial, são retirados desta solução alguns esquemas de corte e tentam-se reagrupar os itens que compõem os mesmos, com o objetivo de minimizar o número de barras padrão necessárias para atender à demanda destes itens. Sendo assim, para cada solução inicial construída, são escolhidos os esquemas de corte que possuam as seguintes características:

a) O esquema com a maior perda individual, ou seja, o esquema que tenha o maior comprimento da perda, considerando que o mesmo possa ser executado mais de uma vez;
b) O esquema que gera a maior perda quando usado, ou seja, o esquema em que o produto de sua perda individual pelo número de vezes que é executado resulta o maior valor;
c) O esquema que contém o maior número de peças;
d) O esquema que gera o maior número de peças quando usado, ou seja, o produto do número de peças do esquema pelo número de vezes que o mesmo é executado.

O objetivo da escolha destes quatro tipos de esquemas é que estão se misturando dois esquemas que geram perdas grandes em relação ao problema, com dois esquemas que contem um maior número de peças. Os dois esquemas que têm o maior número de peças indicam que as peças que componham os mesmos são de menores comprimentos, favorecendo possíveis melhores combinações na construção dos novos esquemas.

Estes esquemas são retirados da solução inicial e transformados em um subproblema e os itens deste subproblema são submetidos a um gerador de padrões viáveis. O gerador de padrões, desenvolvido por Rocha (1997), obtém, por processos combinatórios, todos os padrões ou esquemas de corte viáveis. Embora este processo requeira grande tempo de processamento e memória para armazenamento dos vetores gerados, será utilizado, uma vez que o subproblema tem a característica de ser pequeno.

A modelagem do subproblema apresenta a mesma estrutura da formulação matemática básica do problema de corte, sendo m o número de itens distintos dos esquemas escolhidos, e após a geração dos padrões viáveis, resolve-se então este problema pelo método Simplex, relaxando a integridade de suas variáveis.

Da solução relaxada são utilizados os esquemas com os valores de $x_j \geq 1$ utilizando apenas a parte inteira, anexando-os então aos esquemas resultantes da solução inicial, que será chamada de Solução Melhorada Incompleta. Como só foi utilizada a parte inteira da variável da solução relaxada, resulta-se então, uma lista de itens com demanda não atendida, em relação à Solução Inicial.

Se a soma dos comprimentos dos itens desta lista (comprimento linear) for menor que o comprimento da barra padrão, constrói-se então um esquema juntando-o à Solução Melhorada Incompleta e escreve-se então a Solução Final.

Caso o comprimento linear da lista seja maior do que o comprimento de uma barra padrão ou menor do que duas vezes o comprimento da barra padrão, o próximo passo será tentar encaixar os itens desta lista nos esquemas já existentes, fazendo trocas por itens de comprimento menor, utilizando como complemento a perda destes esquemas. Desta forma o comprimento linear da lista tende a diminuir. Se o comprimento desta lista for reduzido até um comprimento menor ou igual à barra padrão, constrói-se então um esquema juntando-o à Solução Melhorada Incompleta e escreve-se então a Solução Final.

Se o comprimento da lista não se enquadrar nas duas situações descritas acima, ou seja, o comprimento linear desta for maior do que duas vezes o comprimento da barra padrão, geram-se então esquemas para atender aos itens da lista, utilizando o procedimento de construção de soluções iniciais da GRASP, anexando estes esquemas à solução inicial.

Sendo assim, serão apresentados a seguir os pseudocódigos que compõem o algoritmo do procedimento de melhoria da solução.

**Procedimento** ALG_MELHORIA ($m, l_1 \ldots l_m, d_1 \ldots d_m, m, S_{inicial}$)

1. Selecionar quatro esquemas da solução inicial $S_{inicial}$;
2. Criar o grupo de esquemas $G_1$ formado pelos esquemas selecionados no passo 1;
3. Executar ALG_TROCAS;

**Fim** ALG_MELHORIA

**Procedimento** ALG_TROCAS ($S_{inicial}, G_1$)

1. Separar $S_{inicial}$ em dois grupos de esquemas: $G_1$ e $G_2 = S_{inicial} - G_1$;
2. Transformar o grupo $G_1$ no subproblema $P_1$;
3. Resolver pelo Simplex o problema $P_1$, relaxando a integridade das variáveis, gerando a solução S';
4. Criar o grupo de esquemas $G_1$', utilizando os esquemas gerados na solução S' que atendam a condição $\lfloor x_j \rfloor > 0$;
5. Criar a solução $S_m$, acrescentando os esquemas do grupo $G_1$' ao grupo $G_2$;
6. Criar lista de itens LIST com demanda não atendida: LIST = $S_{inicial} - S_m$;
7. Calcular o comprimento linear CL de LIST, $CL = \sum_{i=1}^{k} l_i d_i$ onde k é o número de itens da lista;
8. Se $CL \leq L$, gerar um esquema de corte e anexar a solução $S_m$ e ir para o passo 23;
9. Se $L \leq CL < 2L$, ir para o passo 12;
10. Se $CL > 2L$, executar ALG_SOLUÇÃO INICIAL para solução da LIST;
11. Acrescentar os esquemas da solução do passo 10 a solução $S_m$ e ir para o passo 23;

12. Selecionar o esquema de corte $ESQ_1$ de $S_m$ com maior perda;
13. Fazer $ESQ_{troca} = ESQ_1$;
14. Decrementar em uma unidade o número de vezes que deve ser executado o $ESQ_1$;
15. Caso número de vezes que deve ser executado $ESQ_1$ tornar zero, excluir $ESQ_1$ da solução $S_m$;
16. Somar o maior item $L_{maior}$ de $ESQ_{troca}$ com a perda $P_{ESQtroca}$ do esquema $ESQ_{troca}$, $L_{troca} = L_{maior} + P_{ESQtroca}$;
17. Selecionar o maior item $Lt_i$ de LIST, decrementando a demanda deste item;
18. Fazer $L_{acumula} = Lt_i$;
19. Se $L_{maior} < L_{acumula} \leq L_{troca}$, trocar $Lt_i$ por $L_{maior}$ no esquema $ESQ_{troca}$ e ir para o passo 22;
20. Se $L_{acumula} < L_{maior}$, somar mais um item ao $L_{acumula}$, $L_{acumula} = L_{acumula} + Lt_{i+1}$, ir para passo 19;
21. Se $L_{acumula} > L_{troca}$, selecionar próximo item da lista $Lt_i = Lt_{i+1}$ e voltar ao passo 19;
22. Calcular novo comprimento linear de LIST e voltar ao passo 8;
23. Guardar Solução Melhorada, $S_{melhor} = S_m$;
24. Calcular o número de barras padrão usadas, $BarrasUsadas_{melhor} = \sum_{j=1}^{n} x_j$;
25. Calcular a perda total, $PerdaTotal_{melhor} = BarrasUsadas_{melhor} \cdot BarraPadrão - \sum_{i=1}^{m} l_i d_i$

**Fim** ALG_TROCAS

Os algoritmos apresentados nas próximas seções receberam as denominações de ALG HB1 e ALG HB2. Estes foram desenvolvidos utilizando parte do algoritmo GRASP-1D (procedimento de construção de soluções iniciais) junto a algoritmos para tratamento de algumas das soluções geradas, visando uma solução final com o menor número de barras padrão necessárias para atender à demanda do problema.

## 5. Algoritmo ALG HB1

Após originar um determinado número de soluções pela fase de construção da GRASP, número este definido pelo número de iterações adotado, escolhem-se as $S_e$ melhores soluções que sejam diferentes.

O número de soluções selecionadas $S_e$ para tratamento é um dado arbitrado, pois quanto maior o número de soluções selecionadas, maior será o tempo de processamento, e maior será a memória para arquivar todas as variáveis em questão. Ao arbitrar nos testes o número de cinco esquemas selecionados, tem-se o tamanho suficiente da amostra com o desígnio de verificar o comportamento do algoritmo de refinamento em função das variações dos arranjos dos itens nos esquemas.

Cada solução selecionada recebe um tratamento independente e finalmente registra-se a melhor solução final entre estas soluções. Este tratamento inicia-se com a seleção dos esquemas e os critérios adotados para escolha destes esquemas foram os mesmos apresentados na seção 4.2.

Em seguida aplica-se o procedimento ALG TROCAS, descrito também na seção 4.2. Recalcula-se a perda porcentual total da solução melhorada, em seguida verifica-se se houve redução da perda total em relação à perda antes do tratamento de melhoria, caso tenha ocorrido redução, aplica-se o procedimento de melhoria novamente a esta nova solução, e calcula-se a nova perda total. Isto deve ser repetido até que não ocorra redução da perda total da solução melhorada, em relação à anterior, registrando esta solução final.

Trata-se então a próxima solução selecionada e compara-se a perda total desta com a perda total da solução anterior, registrando então sempre a melhor das soluções tratadas.

Desta maneira, o algoritmo ALG HB1 é descrito como segue.

**Procedimento** ALG HB1 ($m, l_1 \ldots l_m, d_1 \ldots d_m, \alpha, L, n_{iterações}, S_e$)

1. Executar ALG_SOLUÇÃO INICIAL até iteração = $n_{iteração}$;
2. Registrar as $S_e$ melhores soluções diferentes: $S_{inicial1}$, $S_{inicial2}$, ...., $S_{inicialSe}$;

3. Atribuir i = 1;
4. Atribuir $S_{final} = S_{inicial1}$;
5. Se i = $S_e$, ir para o passo 19;
6. Ler $S_{iniciali}$;
7. Atribuir $S_{inicial} = S_{iniciali}$;
8. Selecionar quatro esquemas da solução inicial $S_{inicial}$;
9. Criar o grupo de esquemas $G_1$ formado pelos esquemas selecionados no passo 8;
10. Executar ALG_TROCAS;
11. Ler saída de ALG_TROCAS ($S_{melhor}$, $BarrasUsadas_{melhor}$, $PerdaTotal_{melhor}$);
12. Se $PerdaTotal_{melhor} < PerdaTotal_i$, ir para o passo 13, senão ir para o passo 15;
13. Atribuir $PerdaTotal_i = PerdaTotal_{melhor}$;
14. Voltar ao passo 8;
15. Atribuir $S_{melhori} = S_{melhor}$;
16. Se $S_{melhori}$ for melhor que $S_{final}$, atribuir $S_{final} = S_{melhori}$;
17. Fazer i = i + 1;
18. Voltar ao passo 5;
19. Escrever ($Solução_{final}$, $PerdaTotal_{final}$, $BarrasUsadas_{final}$);

**Fim** ALG HB1

## 6. Algoritmo ALG HB2

Este algoritmo basicamente segue os mesmos procedimentos do ALG HB1, só diferindo na escolha dos esquemas que serão selecionados no início do tratamento das soluções iniciais.

Neste algoritmo são escolhidos os esquemas de corte que possuam as seguintes características:

a) Os esquemas de corte que têm o comprimento da perda no esquema maior ou igual ao comprimento que representa a perda total porcentual da solução inicial na barra padrão;
b) O esquema que contém o maior número de peças;
c) O esquema que gera o maior número de peças quando usado, ou seja, o produto do número de peças do esquema pelo número de vezes que o mesmo é executado.

O motivo de incluir os esquemas descritos nas alíneas b e c sugere que, esquemas que têm o maior número de peças, indicam que as peças que componham os mesmos são de menores comprimentos, favorecendo então, possíveis melhores combinações na construção dos novos esquemas.

O algoritmo ALG HB2 é descrito como segue.

**Procedimento** ALG HB2 ($m, l_1 \ldots l_m, d_1 \ldots d_m, \alpha, L, n_{iterações}, S_e$)

1. Executar ALG_SOLUÇÃO INICIAL até iteração = $n_{iteração}$;
2. Registrar as $S_e$ melhores soluções diferentes: $S_{inicial1}$, $S_{inicial2}$, ...., $S_{inicialSe}$;
3. Atribuir i = 1;
4. Atribuir $S_{final} = S_{inicial1}$;
5. Se i = $S_e$, ir para o passo 22;
6. Ler $S_{iniciali}$;
7. Atribuir $S_{inicial} = S_{iniciali}$;

8. Calcular a perda total relativa de $S_{inicial}$, $PerdaRelativa_i = Perda_{total} / BarrasUsadas_{inicial} \cdot L$;
9. Calcular o comprimento equivalente $C_e$ da $PerdaRelativa_i$ em uma barra padrão, $C_e = L \cdot PerdaRelativa_i$;
10. Selecionar os esquemas da solução inicial $S_{inicial}$, $Perda_{esq} > C_e$;
11. Criar o grupo de esquemas $G_1$ formado pelos esquemas selecionados no passo 10;
12. Executar ALG_TROCAS;
13. Ler saída de ALG_TROCAS ($S_{melhor}$, $BarrasUsadas_{melhor}$, $PerdaTotal_{melhor}$);
14. Calcular $PerdaRelativa_{melhor}$, $PerdaRelativa_{melhor} = PerdaTotal_{melhor} / BarrasUsadas_{melhor} \cdot L$;
15. Se $PerdaRelativa_{melhor} < PerdaRelativa_i$, ir para o passo 16, senão ir para o passo 18;
16. Atribuir $PerdaRelativa_i = PerdaRelativa_{melhor}$;
17. Voltar ao passo 9;
18. Atribuir $S_{melhori} = S_{melhor}$;
19. Se $S_{melhori}$ for melhor que $S_{final}$, atribuir $S_{final} = S_{melhori}$;
20. Fazer i = i + 1;
21. Voltar ao passo 5;
22. Escrever ($Solução_{final}$, $Perda_{total}$, $BarrasUsadas_{total}$);

**Fim** ALG HB2

## 7. Testes Computacionais

Para observar o comportamento de algoritmos híbridos a partir da metaheurística GRASP, foram implementados os algoritmo GRASP 1D, ALG HB1 e ALG HB2, apresentados nas seções 4, 5 e 6, respectivamente, e executados testes com instâncias práticas retiradas de fábricas.

Estas instâncias foram divididas em quatro grupos (Bg1, Bg2, Bg3 e Bg4) de acordo com dados dos projetos. As instâncias não seguem uma sequência padronizada de numeração, pois foi seguida a codificação original. Por exemplo, no grupo Bg1, que vai de Bg1001 até Bg1010, falta a instância Bg1007. Estas instâncias foram excluídas por apresentarem dados incompletos ou incoerentes.

Em função dos resultados dos testes apresentados por Vieira Neto, Velasco e Paula Junior (2010), optou-se por testar cada instância, rodando o procedimento de construção da solução da GRASP com o parâmetro $\alpha$ variando de 0,5 a 1,0 com incremento de 0,05, uma vez que a solução com o parâmetro $\alpha$ abaixo de 0,5, na maioria dos casos não leva, na fase de construção da solução, aos melhores resultados. Para cada solução obtida com determinado valor de $\alpha$, foram utilizadas 100 iterações.

Após os testes realizados com a heurística FFD e os algoritmos híbridos apresentados GRASP-1D, ALG HB1 e ALG HB2, reuniram-se, nas tabelas 1, 2, 3 e 4, as perdas resultantes da aplicação desses métodos para as instâncias práticas testadas com o objetivo de avaliar o comportamento comparativamente de cada um destes métodos, quando aplicados ao problema de corte unidimensional. Para se ter uma visão global destes resultados, os mesmos foram representados também na figura 2.

No processamento das soluções utilizando o algoritmo ALG HB2 apresentou um aumento no tempo para os problemas maiores, com relação ao algoritmo ALG HB1, explicado pelo maior número de itens que compõe o grupo G1, criado no passo 11 do algoritmo ALG HB2. Neste algoritmo, diferente do ALG HB1 que, selecionam apenas quatro esquemas para o grupo

G1, selecionam-se esquemas que atendem a condição do passo 10 que, na maioria dos casos resulta em mais de quatro esquemas que, conseqüentemente gera um maior número de itens a serem combinados. Este tempo que é necessário principalmente para gerar os padrões viáveis e executar o Simplex.

TABELA 1: Comparativo das perdas percentuais das instâncias do grupo BG 1.

| Código | m | Simplex | FFD | | GRASP-1D | | | | ALG HB1 | | | | ALG HB2 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | total barras | total barra | perda (%) | total barra | perda (%) | t (s) | red (%) | total barra | perda (%) | t (s) | red (%) | total barra | perda (%) | t (s) | red (%) |
| Bg 1001 | 8 | 185.99 | 192 | 3.51 | 188 | 1.45 | 7 | 58.69 | 187 | 0.93 | 8 | 73.50 | 187 | 0.93 | 6 | 73.50 |
| Bg 1002 | 14 | 209.70 | 216 | 4.34 | 212 | 2.54 | 8 | 41.47 | 212 | 2.54 | 5 | 41.47 | 210 | 1.61 | 6 | 62.90 |
| Bg 1003 | 15 | 206.97 | 213 | 2.83 | 210 | 1.44 | 7 | 49.12 | 209 | 0.97 | 6 | 65.72 | 209 | 0.97 | 4 | 65.72 |
| Bg 1004 | 9 | 54.95 | 57 | 5.00 | 55 | 1.55 | 8 | 69.00 | 55 | 1.55 | 4 | 69.00 | 55 | 1.55 | 4 | 69.00 |
| Bg 1005 | 18 | 181.05 | 186 | 3.43 | 184 | 2.38 | 7 | 30.61 | 184 | 2.38 | 6 | 30.61 | 182 | 1.31 | 3 | 61.81 |
| Bg 1006 | 8 | 145.50 | 147 | 3.86 | 147 | 3.86 | 7 | 0.00 | 147 | 3.86 | 3 | 0.00 | 146 | 3.20 | 3 | 17.10 |
| Bg 1008 | 12 | 77.85 | 81 | 4.56 | 80 | 3.37 | 7 | 26.10 | 79 | 2.15 | 5 | 52.85 | 79 | 2.15 | 3 | 52.85 |
| Bg 1009 | 10 | 30.00 | 32 | 9.80 | 31 | 6.89 | 7 | 29.69 | 31 | 6.89 | 4 | 29.69 | 30 | 3.78 | 6 | 61.43 |
| Bg 1010 | 21 | 291.78 | 301 | 3.11 | 296 | 1.47 | 7 | 52.73 | 295 | 1.14 | 8 | 63.34 | 295 | 1.14 | 6 | 63.34 |

TABELA 2: Comparativo das perdas percentuais das instâncias do grupo BG 2.

| Código | m | Simplex | FFD | | GRASP-1D | | | | ALG HB1 | | | | ALG HB2 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | total barras | total barra | perda (%) | total barra | perda (%) | t (s) | red (%) | total barra | perda (%) | t (s) | red (%) | total barra | perda (%) | t (s) | red (%) |
| Bg 2001 | 28 | 1522.81 | 1608 | 8.12 | 1575 | 6.20 | 7 | 23.65 | 1550 | 4.68 | 11 | 42.36 | 1528 | 3.31 | 13 | 59.24 |
| Bg 2002 | 46 | 1458.69 | 1562 | 7.01 | 1525 | 4.76 | 7 | 32.10 | 1509 | 3.75 | 11 | 46.50 | 1485 | 2.19 | 15 | 68.76 |
| Bg 2003 | 23 | 671.85 | 680 | 1.37 | 679 | 1.23 | 8 | 10.22 | 679 | 1.23 | 4 | 10.22 | 679 | 1.23 | 4 | 10.22 |
| Bg 2004 | 23 | 349.67 | 365 | 4.71 | 362 | 3.92 | 8 | 16.77 | 359 | 3.11 | 6 | 33.97 | 355 | 2.02 | 9 | 57.11 |
| Bg 2005 | 23 | 340.29 | 346 | 4.78 | 344 | 4.22 | 7 | 11.72 | 344 | 4.22 | 5 | 11.72 | 342 | 3.66 | 7 | 23.43 |
| Bg 2006 | 23 | 695.25 | 715 | 4.55 | 710 | 3.88 | 7 | 14.73 | 710 | 3.88 | 7 | 14.73 | 699 | 2.37 | 9 | 47.91 |
| Bg 2007 | 23 | 365.57 | 371 | 2.49 | 368 | 1.69 | 7 | 32.13 | 367 | 1.43 | 5 | 42.57 | 367 | 1.43 | 6 | 42.57 |
| Bg 2009 | 46 | 1905.97 | 1954 | 2.49 | 1940 | 1.78 | 8 | 28.51 | 1939 | 1.73 | 6 | 30.52 | 1934 | 1.48 | 7 | 40.56 |
| Bg 2010 | 23 | 731.13 | 736 | 2.16 | 734 | 1.89 | 8 | 12.50 | 734 | 1.89 | 5 | 12.50 | 732 | 1.63 | 6 | 24.54 |

TABELA 3: Comparativo das perdas percentuais das instâncias do grupo BG 3.

| Código | m | Simplex | FFD | | GRASP-1D | | | | ALG HB1 | | | | ALG HB2 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | total barras | total barra | perda (%) | total barra | perda (%) | t (s) | red (%) | total barra | perda (%) | t (s) | red (%) | total barra | perda (%) | t (s) | red (%) |
| Bg 3001 | 49 | 2023.46 | 2113 | 4.49 | 2107 | 4.22 | 7 | 6.01 | 2103 | 4.04 | 10 | 10.02 | 2050 | 1.56 | 16 | 65.26 |
| Bg 3002 | 68 | 1987.01 | 2088 | 5.03 | 2075 | 4.43 | 8 | 11.93 | 2066 | 4.02 | 11 | 20.08 | 2020 | 1.83 | 40 | 63.62 |
| Bg 3004 | 37 | 529.92 | 551 | 4.05 | 547 | 3.35 | 8 | 17.28 | 547 | 3.35 | 6 | 17.28 | 544 | 2.81 | 6 | 30.62 |
| Bg 3005 | 34 | 510.63 | 534 | 4.43 | 531 | 3.89 | 7 | 12.19 | 526 | 2.97 | 11 | 32.96 | 523 | 2.42 | 9 | 45.37 |
| Bg 3007 | 34 | 551.53 | 563 | 7.21 | 560 | 6.71 | 8 | 6.93 | 560 | 6.71 | 6 | 6.93 | 555 | 5.87 | 5 | 18.59 |
| Bg 3009 | 68 | 2842.92 | 2882 | 1.34 | 2880 | 1.27 | 9 | 5.22 | 2880 | 1.27 | 6 | 5.22 | 2874 | 1.07 | 8 | 20.15 |
| Bg 3010 | 37 | 1084.52 | 1112 | 2.71 | 1111 | 2.62 | 8 | 3.32 | 1108 | 2.36 | 6 | 12.92 | 1106 | 2.18 | 5 | 19.56 |
| Bg 3011 | 14 | 92.69 | 95 | 4.8 | 94 | 3.79 | 7 | 21.04 | 94 | 3.79 | 4 | 21.04 | 94 | 3.79 | 3 | 21.04 |

TABELA 4: Comparativo das perdas percentuais das instâncias do grupo BG 4.

| Código | m | Simplex | FFD | | GRASP-1D | | | | ALG HB1 | | | | ALG HB2 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | total barras | total barra | perda (%) | total barra | perda (%) | t (s) | red (%) | total barra | perda (%) | t (s) | red (%) | total barra | perda (%) | t (s) | red (%) |
| Bg 4001 | 67 | 2165.31 | 2273 | 4.89 | 2268 | 4.68 | 8 | 4.29 | 2060 | 4.34 | 11 | 11.25 | 2194 | 1.47 | 25 | 69.94 |
| Bg 4002 | 79 | 2124.62 | 2236 | 5.04 | 2230 | 4.78 | 8 | 5.16 | 2216 | 4.18 | 14 | 17.06 | 2168 | 2.06 | 25 | 59.13 |
| Bg 4003 | 43 | 1004.77 | 1018 | 1.33 | 1014 | 0.95 | 11 | 28.57 | 1014 | 0.95 | 5 | 28.57 | 1012 | 0.75 | 20 | 43.61 |
| Bg 4004 | 47 | 594.13 | 617 | 3.75 | 614 | 3.28 | 8 | 12.53 | 612 | 2.96 | 6 | 21.07 | 606 | 2.00 | 6 | 46.67 |
| Bg 4005 | 44 | 575.54 | 601 | 4.24 | 597 | 3.60 | 7 | 15.09 | 594 | 3.11 | 12 | 26.65 | 589 | 2.29 | 10 | 45.99 |
| Bg 4006 | 44 | 941.68 | 980 | 3.96 | 977 | 3.66 | 7 | 7.58 | 965 | 2.46 | 13 | 37.88 | 953 | 1.24 | 16 | 68.69 |
| Bg 4007 | 44 | 559.32 | 565 | 1.73 | 565 | 1.73 | 8 | 0.00 | 565 | 1.73 | 6 | 0.00 | 563 | 1.38 | 8 | 20.23 |
| Bg 4010 | 47 | 1212.92 | 1245 | 2.63 | 1244 | 2.55 | 8 | 3.04 | 1241 | 2.32 | 6 | 11.79 | 1240 | 2.24 | 5 | 14.83 |
| Bg 4011 | 24 | 126.02 | 129 | 4.63 | 128 | 3.88 | 8 | 16.20 | 127 | 3.13 | 4 | 32.40 | 127 | 3.13 | 4 | 32.40 |

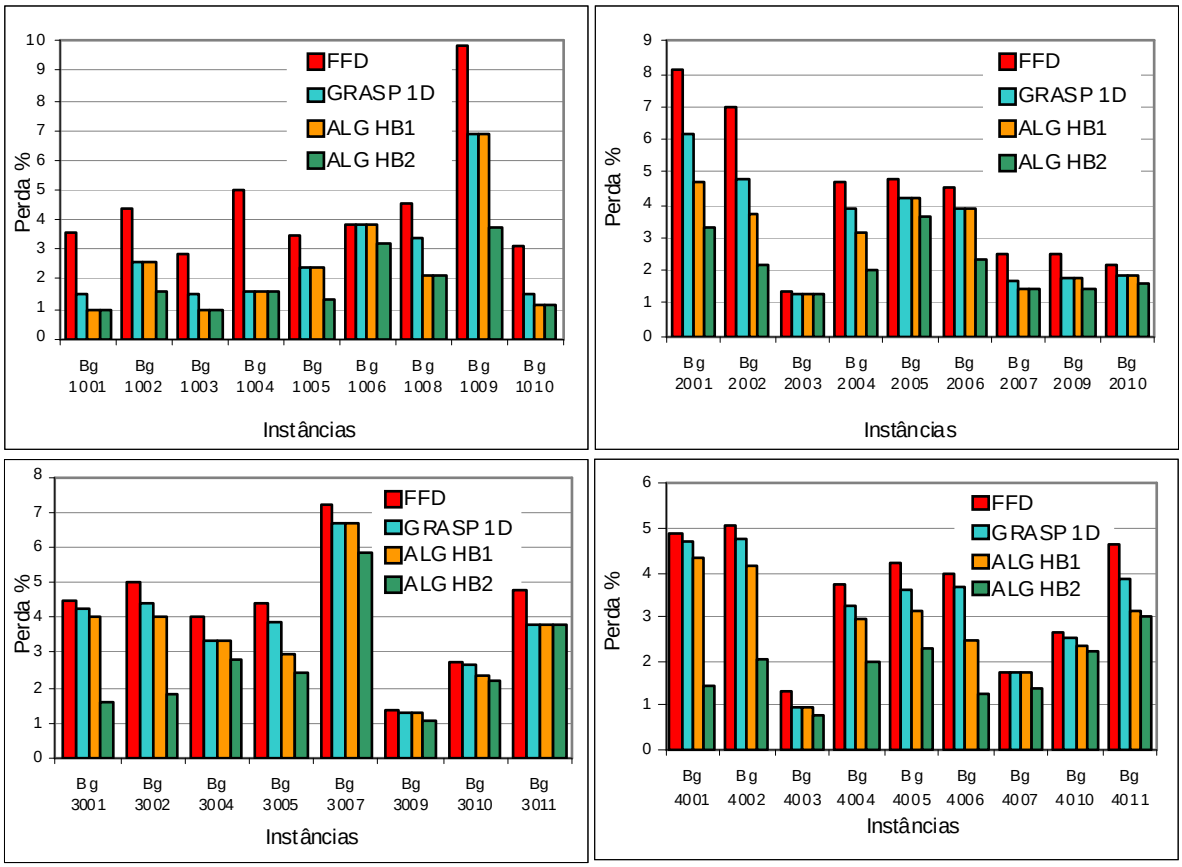


FIGURA 2 - Gráficos comparativos de soluções das instâncias.

Observa-se que em apenas dois casos a GRASP-1D e o ALG HB1 não conseguiram soluções melhores do que a FFD, já o ALG HB2 conseguiu em todos os casos as melhores soluções para todos os casos.

Não se pode avaliar com precisão, o quanto próximo, os resultados finais para estas instâncias estão da solução ótima, uma vez que não se dispõe destes valores. Porém observando o total de barras necessário para a solução pelo método Simplex relaxado, observa-se em alguns casos que a solução ótima foi alcançada (Bg1002, Bg1004, Bg1005, Bg1006, Bg1009, Bg2010). Esta afirmação pode ser feita, pois o resultado destas instâncias é o menor inteiro maior que a solução do Simplex. Em outros casos pela proximidade do resultado inteiro, do resultado da solução relaxada, leva a crer que seria a solução ótima ou estaria bem próxima da mesma.

## 8. Conclusões

A filosofia da busca local, utilizada nos três algoritmos híbridos apresentados, com pequenas modificações para cada um deles, mostrou-se de grande eficiência para o tratamento de soluções iniciais, isto fica evidente quando se observa o percentual de redução que cada método alcançou com relação à heurística FFD.

O algoritmo ALG HB2 mostrou um desempenho muito bom, conseguindo alcançar a solução ótima para algumas instâncias e para as outras, se não for a solução ótima, está em torno de 1% desta.

A fase do procedimento de melhoria, que requer um maior tempo, é a fase de gerar os padrões viáveis para solução relaxada do subproblema. Em nenhum caso este tempo passou de 0,5 segundo, pois o critério de escolha dos quatro esquemas que irão gerar o subproblema produz um número pequeno de itens e comprimentos dos mesmos, que facilitam a geração de padrões viáveis.

Outros estudos podem ser realizados, explorando ainda mais procedimentos de melhoria, modificando a seleção dos esquemas para compor o subproblema, bem como desenvolver outros algoritmos para o reagrupamento dos itens do subproblema.

## Referências Bibliográficas


**Cintra, G.F.** (1998). *Algoritmos Híbridos para Problemas de Corte Unidimensional*. Dissertação (Mestrado em Matemática Aplicada) - Instituto de Matemática e Estatística - USP, São Paulo-SP.

**Cui, Y., Zhao, X., Yang, Y. and Yu, P.** (2008). A heuristic for the one dimensional cutting stock problem with pattern reduction. *Proceedings of the Institution of Mechanical Engineers, Part B, Journal of Engineering Manufacture*, v.222, n.6, p.677-685.

**Dyckhoff, H.** (1990). A Typology of Cutting and Packing Problems. *European Journal of Operational Research*, v.44, pp.145–159.

**Feo, T. A., Resende, M. G. C.** (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, v.6, p.109-133.

**Foerster, H. and Wäscher, G.** (2000). Pattern reduction in one-dimensional cutting stock problem. *International Journal of Production Research*, v.38, n.7, p.1657-1676.

**Gilmore, P.; Gomory, R.** (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, v.9, p.849-859.

**Haessler, R. W.** (1975). Controlling cutting pattern changes in one-dimensional trim problems. *Operations Research*, v.23, n.3, p.483-493.

**McDiarmid, C.** (1999). Pattern minimisation in cutting stock problems. *Discrete Applied Mathematics*, v. 98, p. 121-130.

**Rocha, M.N.** (1997). *Otimização de Cortes de Barras*. Dissertação (Mestrado em Ciências da Computação) - Programa de Pós-Graduação em Ciência da Computação - UFMG, Belo Horizonte-MG.

**Stadtler, H.** (1990). A one-dimensional cutting stock problem in the aluminum industry and its solution. *European Journal of Operational Research*, v.44, p.209-223.

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

**Vieira Neto, E., Velasco, A.S. e Paula Junior, G.G.** (2010). GRASP: a influência da lista restrita de candidatos nas soluções iniciais para otimização do corte unidimensional. *Gepros - Gestão da Produção, Operações e Sistemas*, v.5, n.3, p.29-44.

**Yanasse, H. H. e Cerqueira, G. R. L.** (2009). Uma heurística baseada em geração sequencial de padrões para o problema de corte de estoque unidimensional com o número reduzido de padrões. *Gestão & Produção*, v.16, n.2, p.200-208.

**Yanasse, H. H. e Limeira, M. S.** (2006). A hybrid heuristic to reduce the number of different patterns in cutting stock problems. *Computers & Operations Research*, v.33, p.2744-2756.