

Problema de Corte de Estoque Unidimensional com Sobras Aproveitáveis: Solução via Metaheurística GRASP

A. R. RIOS¹ e K. C. POLDI^{2*}

Recebido em 3 de fevereiro de 2020 / Aceito em 9 de julho de 2020

RESUMO. Neste artigo apresentamos um estudo sobre o Problema de Corte de Estoque com Sobras Aproveitáveis (PCESA) unidimensional. O Problema de Corte de Estoque (PCE) consiste em cortar um conjunto de objetos, disponíveis em estoque, para produzir um conjunto de itens em quantidades e tamanhos especificados. O PCESA é um PCE no qual nem todas as sobras de um processo de corte são consideradas perdas, ou seja, quando uma sobra for maior que um determinado comprimento, ela retorna ao estoque para ser usada nos processos de corte subsequentes. Assim, o objetivo do problema abordado é otimizar os planos de corte a serem aplicados em objetos padronizados e objetos não padronizados (sobras de cortes anteriores); atender a demanda conhecida de itens menores, de forma a minimizar a perda de material, controlando a quantidade de retalhos gerados. Para resolver este problema, duas heurísticas e três versões da metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP) foram desenvolvidas. Testes computacionais foram realizados para os procedimentos propostos em instâncias da literatura. As abordagens propostas para a solução apresentaram boas soluções em relação ao número de sobras geradas.

Palavras-chave: problema de corte de estoque unidimensional, sobras aproveitáveis, metaheurística, GRASP.

1 INTRODUÇÃO

O Problema de Corte de Estoque com Sobras Aproveitáveis (PCESA) unidimensional consiste em cortar objetos disponíveis em estoque para produzir itens demandados, otimizando uma determinada função objetivo, na qual apenas uma dimensão é relevante no processo de corte, onde uma sobra pode ser classificada como perda ou retalho. Sendo retalho, ele retorna ao estoque para ser usado no processo de corte subsequente.

Muitas indústrias, em seus processos de corte, reutilizam as sobras para atender às demandas futuras. O primeiro artigo que se refere ao problema de corte de estoque com as sobras aproveitáveis

*Autor correspondente: Kelly Cristina Poldi – E-mail: kelly@ime.unicamp.br

¹Departamento de Matemática Aplicada, IMECC, Universidade Estadual de Campinas, R. Sérgio Buarque de Holanda, 651, 13083-859, Campinas, SP, Brasil – E-mail: ariqule@hotmail.com <https://orcid.org/0000-0001-9596-4125>

²Departamento de Matemática Aplicada, IMECC, Universidade Estadual de Campinas, R. Sérgio Buarque de Holanda, 651, 13083-859, Campinas, SP, Brasil – E-mail: kelly@ime.unicamp.br <https://orcid.org/0000-0002-1649-6843>

encontradas na literatura foi proposto por Brown em 1971 [3], mas somente posteriormente, esse problema foi abordado diretamente. Algumas abordagens de solução para PCESA podem ser encontradas em, Roodman [14], Scheithauer [15], Gradisar et al. [9], Abuabara e Morabito [1], Nicola [12], Cherri et al. [4], Koch et al. [10], Cherri et al. [5], Arenales et al. [2] e entre outros.

Em 2014, Cherri et al. [6] reuniram e analisaram pesquisas da literatura que consideram o PCESA unidimensional. Em cada artigo analisado, foram apresentadas aplicações para o problema, o modelo matemático quando existente e comentários sobre os resultados obtidos em cada uma deles.

Problemas de corte e empacotamento são NP-difíceis (veja Garey e Johnson [8]), uma vez que não existem algoritmos na literatura para resolver esses tipos de problemas em tempo polinomial. Existem alguns algoritmos exatos, mas a maioria deles tem complexidade de tempo exponencial, resultando em um esforço computacional muito alto. No entanto, existem vários métodos heurísticos de resolução.

Algumas metaheurísticas conhecidas na literatura são algoritmo genético, *simulated annealing*, GRASP, busca tabu, colônia de formigas, colônia de abelhas, entre outras (Resende e Sousa [13]).

Neste artigo, abordamos o PCESA unidimensional e propomos duas heurísticas e três metaheurísticas GRASP como método de resolução, sendo que, as duas heurísticas são utilizadas para iniciar as três metaheurísticas. A metaheurística GRASP foi proposta por Feo e Resende em 1995 [7] e é um procedimento iterativo composto por duas fases: construção da solução e busca local na vizinhança da solução encontrada. No fim do procedimento, a melhor solução entre todas as iterações é a solução para o problema, com a melhor solução sendo armazenada em cada iteração e a pior sendo descartada.

Em 2013, Neto et al. [11] considerou o PCE clássico unidimensional, sem considerar as sobras, com a abordagem de solução sendo a metaheurística GRASP. Com base em Neto et al. [11], desenvolvemos três metaheurísticas GRASP para a resolução do PCESA unidimensional.

2 O PROBLEMA DE CORTE DE ESTOQUE UNIDIMENSIONAL COM SOBRAS APROVEITÁVEIS

Um Problema de Corte de Estoque (PCE) unidimensional, ou seja, apenas uma dimensão do objeto é relevante, consiste em cortar objetos de comprimento L_k , com $k = 1, \dots, \bar{K}$ em itens menores de comprimento ℓ_i , com $i = 1, \dots, m$, de forma a atender a demanda conhecida *a priori* $d_i, i = 1, \dots, m$. Em um PCE clássico, o objetivo pode ser minimizar o total de objetos cortados, minimizar perda de material ou minimizar custo de material.

Nesse processo de cortar peças menores a partir de objetos maiores pode ocorrer sobras de material que podem ou não ser descartadas. Problemas como esses, são conhecidos na literatura como problema de corte de estoque unidimensional com sobras de material aproveitáveis.

Definição 2.1. Definimos *sobra* como sendo uma peça cortada, que não é um item demandado.

Definição 2.2. Definimos *retalho* como sendo uma peça cortada, que não é um item demandado, grande o suficiente para ser reutilizado.

O comprimento mínimo para uma sobra ser considerada retalho pode ser, por exemplo, o comprimento do menor item ou a média dos comprimentos dos itens demandados. No PCESA, uma sobra que não seja nula e que não seja considerada um retalho é considerada como perda.

Definição 2.3. Definimos *perda* como sendo um pedaço cortado que não seja um item demandado, e que não possa ser reaproveitado.

Para melhor compreensão considere o exemplo a seguir, em que os dados de estoque e demanda estão representados na Figura 1, e o comprimento mínimo para uma sobra ser considerada retalho (δ) é todo comprimento de sobra que for maior ou igual ao menor comprimento de item demandado, ou seja, se o comprimento da sobra for maior ou igual a quatro, obtemos um retalho.

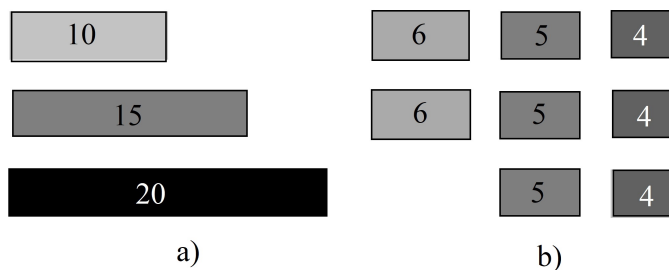


Figura 1: a) Objetos em estoque; b) Itens demandados.

Neste exemplo, a Figura 1a apresenta três tipos de objetos de comprimentos $L_1 = 10$, $L_2 = 15$ e $L_3 = 20$, cada um com uma unidade em estoque, enquanto a Figura 1b apresenta três tipos de itens de comprimentos $\ell_1 = 6$, $\ell_2 = 5$ e $\ell_3 = 4$ com as respectivas demandas $d_1 = 2$, $d_2 = 3$ e $d_3 = 3$. A Figura 2 apresenta três soluções possíveis para o exemplo, em que:

- Figura 2a ilustra uma solução que resulta em um retalho de comprimento 6;
- Figura 2b ilustra uma solução que resulta em duas perdas com comprimentos 3 cada;
- Figura 2c ilustra uma solução que resulta em um retalho de comprimento 4 e uma perda de comprimento 2.

Neste estudo, os itens não devem ser produzidos em excesso; portanto, quando o comprimento de uma sobra é igual ao comprimento de um item demandado, cuja demanda já foi atendida, ela é considerada um retalho.

Não há um modelo matemático na literatura que trate exatamente o caso que abordamos neste artigo. Há alguns trabalhos publicados que tentam modelar o problema (Scheithauer [15], Gradišar et al. [9], Abuabara e Morabito [1], Koch et al. [10], Arenales et al. [2]), mas recaem em

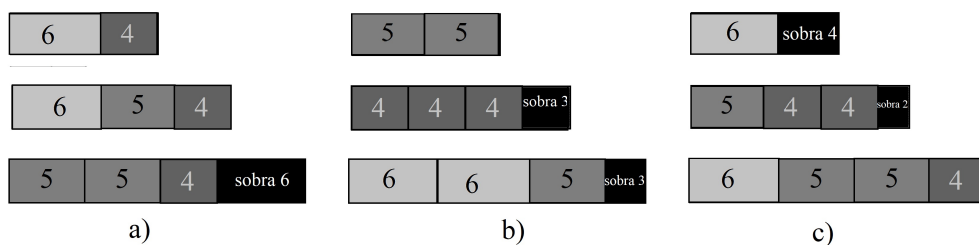


Figura 2: Possíveis soluções para o exemplo de problema de corte.

casos específicos. Por exemplo, alguns modelos se baseiam em discretização dos tamanhos das sobras (Arenales et al. [2]), ou seja, perdem a generalidade pois pode haver algum comprimento de sobra ótimo que não foi considerado.

Na próxima seção, abordamos os procedimentos heurísticos e metaheurísticos para a resolução do PCESA.

3 PROCEDIMENTOS HEURÍSTICOS E METAHEURÍSTICOS PARA RESOLVER O PCESA UNIDIMENSIONAL

Nesta seção, apresentamos dois procedimentos heurísticos para determinar padrões de corte a partir de objetos não padronizados (retalhos resultantes de cortes anteriores) e três metaheurísticas GRASP para obter padrões de corte de objetos padronizados (comprados de fornecedores) com o objetivo de resolver o PCESA unidimensional.

Para os procedimentos propostos, as seguintes informações são consideradas:

- \bar{K} : quantidade de objetos padronizados e não padronizados no estoque inicial;
- K : quantidade de objetos não padronizados no estoque inicial;
- m : quantidade de tipos de itens demandados;
- ℓ_i : comprimento do item tipo i , com $i = 1, \dots, m$;
- d_i : demanda do item tipo i , com $i = 1, \dots, m$;
- L_k : comprimento do objeto tipo k , com $k = 1, \dots, \bar{K}$;
- e_k : disponibilidade do objeto tipo k , com $k = 1, \dots, \bar{K}$;

- $\delta = \frac{\sum_{i=1}^m \ell_i}{m}$: comprimento mínimo para uma sobra ser considerada retalho;

- $\varepsilon = \frac{\max\{\ell_i\} - \min\{\ell_i\}}{m}$, $i = 1, \dots, m$: comprimento máximo para uma perda ser considerada aceitável, isto é, um padrão de corte com perda maior que ε e menor que δ não é aceito para fazer parte da solução.
- $\alpha_1 = 0,5$ e $\alpha_2 = 0,2$: parâmetros de aleatoriedade.
- p_{jk} : comprimento da sobra gerada ao executar o padrão de corte j para o objeto tipo k ;
- x_{jk} : frequência do padrão de corte j para o objeto tipo k ;
- a_{jk} : padrão de corte j para o objeto tipo k ;
- a_{ijk} : quantidade de itens do tipo i no padrão de corte j para o objeto tipo k ;
- n : quantidade total de padrões de corte gerados;
- A_1, A_2 : conjuntos dos padrões de corte gerados para objetos não padronizados (retalhos resultantes de cortes anteriores) e padronizados, respectivamente;
- X : conjunto das frequências dos padrões de corte;
- LRC_i : lista restrita de candidatos composta por itens;
- LRC_p : lista restrita de candidatos composta por padrões de corte;
- P, M : heterogeneidade dos itens em uma determinada classe de instâncias, pouco heterogênea e muito heterogênea, respectivamente;
- k_1, k_2 : quantidade de padrões de corte que produzem retalhos e perdas, respectivamente;
- CPG : comprimento total de perdas geradas;
- RG : quantidade total de retalhos gerados;
- RU : quantidade de retalho no estoque inicial;
- RF : quantidade de retalho no estoque final (após obter plano de corte).

3.1 Heurísticas para obter os padrões de corte para objetos não padronizados

Para obter padrões de corte a partir de objetos não padronizados, foram propostos duas heurísticas, PCR_1 e PCR_2 . Em PCR_1 e PCR_2 , os padrões de corte são obtidos, antes de iniciar a metaheurística GRASP, resolvendo-se um problema da mochila restrito. Para tal, os objetos não padronizados são selecionados para serem usados em ordem crescente de comprimento. Após a obtenção de um padrão de corte, a sobra é analisada, de acordo com um critério adotado, para decidir se o mesmo é aceito na solução. Para cada objeto não padronizado em estoque, apenas um tipo de padrão de corte é construído e é usado o máximo de vezes possível.

Para **PCR₁** utilizamos $p_{jk} \leq \frac{\epsilon}{4}$ e $p_{jk} \cdot x_{jk} > \frac{\epsilon}{2}$. Para **PCR₂** utilizamos $p_{jk} \leq \frac{2\epsilon}{m}$. Esses valores foram escolhidos após realizarmos testes computacionais preliminares, nos quais, para esses valores, pudemos observar melhores resultados com relação ao comprimento das perdas.

No fim de cada Procedimento **PCR₁** e **PCR₂**, é retornada a demanda por itens ainda não atendidos. Essa demanda residual é usada para iniciar a metaheurística GRASP.

A seguir, apresentamos um procedimento, denominado **PCR₁**, para a obtenção de padrões de corte a partir de objetos não padronizados.

Procedimento 1 PCR₁

Entrada: $(m, K, \ell_1, \dots, \ell_m, d_1, \dots, d_m, L_1, \dots, L_K, e_1, \dots, e_K, \epsilon)$

Faça: $j = 0$ e $L_1 \leq L_2 \leq \dots \leq L_K$

1 **Para** $k = 1, \dots, K$, **faça**

2 $j = j + 1$

3 **Se** $L_k \geq \min_{1 \leq i \leq m} \{\ell_i\}$, **então**

4 Obtenha o padrão de corte a_{jk} através da resolução de um problema da mochila restrito e calcule a sobra p_{jk}

5 **Se** $p_{jk} \leq \frac{\epsilon}{4}$, **então**

6 Calcule: $x_{jk} = \min \left\{ e_k, \min_{1 \leq i \leq m} \left\{ \left\lfloor \frac{d_i}{a_{ijk}} \right\rfloor \text{ tal que } a_{ijk} \neq 0 \right\} \right\}$

7 **Enquanto** $p_{jk} \cdot x_{jk} > \frac{\epsilon}{2}$, **faça**

8 | Decremente uma unidade da frequência

9 **Fim**

10 **Fim**

11 **Fim**

12 Atualize: $d_1, \dots, d_m, e_1, \dots, e_K, A_1$ e X

13 **Fim**

14 Calcule: CPG, RU

Saída: $(d_1, \dots, d_m, A_1, X, CPG, RU)$.

O Procedimento 2, denominado **PCR₂**, apresentado a seguir, é uma outra heurística utilizada para determinar padrões de corte a partir de objetos não padronizados.

Procedimento 2 PCR₂**Entrada:** $(m, K, \ell_1, \dots, \ell_m, d_1, \dots, d_m, L_1, \dots, L_K, e_1, \dots, e_K, \varepsilon)$ **Faça:** $j = 0$ e $L_1 \leq L_2 \leq \dots \leq L_K$

- 1 **Para** $k = 1, \dots, K$, **faça**
 - 2 $j = j + 1$
 - 3 **Se** $L_k \geq \min_{1 \leq i \leq m} \{\ell_i\}$, **então**
 - 4 Obtenha o padrão de corte a_{jk} através da resolução de um problema da mochila restrito e calcule a sobra p_{jk}
 - 5 **Se** $p_{jk} \leq \frac{2\varepsilon}{m}$, **então**
 - 6 Calcule: $x_{jk} = \min \left\{ e_k, \min_{1 \leq i \leq m} \left\{ \left\lfloor \frac{d_i}{a_{ijk}} \right\rfloor \text{ tal que } a_{ijk} \neq 0 \right\} \right\}$
 - 7 **Fim**
 - 8 **Fim**
 - 9 Atualize: $d_1, \dots, d_m, e_1, \dots, e_K, A_1$ e X
 - 10 **Fim**
 - 11 Calcule: CPG, RU
- Saída:** $(d_1, \dots, d_m, A_1, X, CPG, RU)$.

3.2 Metaheurística GRASP

A metaheurística GRASP, proposta por Feo e Resende [7], é um procedimento iterativo, no qual cada iteração é composta de duas fases, construção da solução, em que uma solução para o problema é construída e busca local, em que é feito uma busca local na vizinhança dessa solução encontrada a fim de obter um ótimo local. Um pseudocódigo para a metaheurística GRASP é apresentado no Procedimento 3.

Procedimento 3 GRASP**Entrada:** (Dados iniciais do problema)

- 1 **Se** o critério de parada GRASP não estiver satisfeito, **então**
 - 2 Construa uma solução;
 - 3 Faça uma busca local na solução encontrada a fim de melhorá-la;
 - 4 Atualize a solução;
 - 5 **Fim**
- Saída:** (Melhor solução encontrada).

3.3 Metaheurística GRASP para resolução do PCESA unidimensional

Três abordagens metaheurísticas, denominadas $GRASP_1$, $GRASP_2$ e $GRASP_3$, foram desenvolvidas para resolver o PCESA unidimensional após a aplicação das Heurísticas PCR_1 ou PCR_2 . Para as três metaheurísticas propostas, três heurísticas foram desenvolvidas para a fase de construção da solução inicial e duas heurísticas para a fase de melhoria (busca local na vizinhança da solução encontrada). O número máximo de iterações foi utilizado como critério de parada para a metaheurística GRASP. Quanto maior o número de iterações, maior será a chance de obtermos resultados melhores, já que a solução é atualizada sempre que uma solução melhor é encontrada numa determinada iteração. Porém, quanto mais iterações, maior será o tempo de execução, dessa forma adotamos 100 como sendo o número máximo de iterações.

3.3.1 Fase de Construção da solução inicial

Para a fase de construção de uma solução inicial, dois tipos de elementos foram considerados para compor uma LRC, itens e padrões de corte. No caso em que a LRC é composta por itens, duas heurísticas foram desenvolvidas para a fase de construção da solução inicial, e no caso em que a LRC é composta de padrões de corte, uma heurística foi desenvolvida para a fase de construção da solução inicial. Para cada uma das três heurísticas da solução inicial, um padrão de corte é construído para um objeto padronizado do estoque escolhido aleatoriamente.

LRC formada por itens: Nesse caso, para construir um padrão de corte, a LRC é construída cada vez que um item é inserido e é composta por itens cujas demandas ainda não foram atendidas. Ao formar uma LRC, um item é selecionado aleatoriamente entre os itens que a compõem e inserido no padrão de corte. Após a obtenção de um padrão de corte, sua frequência é determinada, a demanda por itens e a disponibilidade de objetos em estoque são atualizadas. Um novo padrão de corte é determinado até que a demanda por itens seja atendida.

Dois heurísticas foram desenvolvidas para construir a solução inicial chamadas CSI_1 e CSI_2 .

Heurística CSI_1 : foram construídos dois tipos de LRC's, chamadas LRC_{i1} , onde α_1 é usado e LRC_{i2} , onde α_2 é usado. Considerando os itens com demanda não atendida, temos: A LRC_{i1} consiste em 50% dos itens de maior comprimento, e a LRC_{i2} consiste em 20% dos itens de menor comprimento.

Heurística CSI_2 : apenas um tipo de LRC foi construída usando α_1 como parâmetro de aleatoriedade. Para a segunda opção de criar uma solução inicial, a LRC é composta por 50% dos itens de maior demanda ainda não atendidos.

O Procedimento 4, denominado CSI_1 , apresenta uma das formas utilizadas para obtenção de uma solução inicial.

Procedimento 4 CSI₁

Entrada: $(m, K, \bar{K}, \ell_1, \dots, \ell_m, d_1, \dots, d_m, L_{K+1}, \dots, L_{\bar{K}}, e_{K+1}, \dots, e_{\bar{K}}, X, \alpha_1, \alpha_2)$

- 1 **Enquanto** a demanda não foi atendida, **faça**
 - 2 **Para** $j = K + 1, \dots, n$, **faça**
 - 3 Selecione aleatoriamente um objeto L_k , para $k = K + 1, \dots, \bar{K}$
 - 4 **Enquanto** o padrão de corte a_{jk} não está finalizado, **faça**
 - 5 Calcule $\beta = \max \{ \ell_i; a_{ijk} = 0, \text{ e } d_i \neq 0 \}$
 - 6 Crie as listas restritas de candidatos:
 - 7 $LRC_{i1} = \{ i; a_{ijk} = 0, d_i \neq 0 \text{ e } \ell_i \geq \alpha_1 \beta \}$
 - 8 $LRC_{i2} = \{ i; a_{ijk} = 0, d_i \neq 0 \text{ e } \ell_i \leq \alpha_2 \beta \}$
 - 9 Selecione aleatoriamente um elemento q de LRC_{i2} (se $id \equiv 2 \pmod{0}$) ou de LRC_{i1} (se $id \not\equiv 2 \pmod{0}$) e insira no padrão de corte a_{jk} quantas vezes forem possíveis
 - 10 **Fim**
 - 11 Calcule: $x_{jk} = \min \left\{ e_k, \min_{1 \leq i \leq m} \left\{ \left\lfloor \frac{d_i}{a_{ijk}} \right\rfloor \text{ tal que } a_{ijk} \neq 0 \right\} \right\}$
 - 12 Atualize: $d_1, \dots, d_m, e_{K+1}, \dots, e_{\bar{K}}, A_2$ e X
 - 13 **Fim**
 - 14 **Fim**
 - 15 Calcule: CPG, RG
- Saída:** $(A_2, X, e_{K+1}, \dots, e_{\bar{K}}, CPG, RG)$

O Procedimento 5, denominado CSI₂, apresenta uma das formas para se obter uma solução inicial.

Procedimento 5 CSI₂

Entrada: $(m, K, \bar{K}, \ell_1, \dots, \ell_m, d_1, \dots, d_m, L_{K+1}, \dots, L_{\bar{K}}, e_{K+1}, \dots, e_{\bar{K}}, X, \alpha_1)$

- 1 **Enquanto** a demanda não foi atendida, **faça**
 - 2 **Para** $j = K + 1, \dots, n$, **faça**
 - 3 Selecione aleatoriamente um objeto L_k , para $k = K + 1, \dots, \bar{K}$
 - 4 **Enquanto** o padrão de corte a_{jk} não está finalizado, **faça**
 - 5 Calcule $\beta = \max \{ d_i; a_{ijk} = 0 \}$
 - 6 Crie a lista restrita de candidatos:
 - 7 $LRC_p = \{ i; a_{ijk} = 0, \text{ e } d_i \neq 0 \text{ e } d_i \geq \alpha_1 \beta \}$ Selecione aleatoriamente um elemento q de LRC_p e insira no padrão de corte a_{jk} quantas vezes forem possíveis
 - 8 **Fim**
 - 9 Calcule: $x_{jk} = \min \left\{ e_k, \min_{1 \leq i \leq m} \left\{ \left\lfloor \frac{d_i}{a_{ijk}} \right\rfloor \text{ tal que } a_{ijk} \neq 0 \right\} \right\}$
 - 10 Atualize: $d_1, \dots, d_m, e_{K+1}, \dots, e_{\bar{K}}, A_2$ e X
 - 11 **Fim**
 - 12 **Fim**
 - 13 Calcule: CPG, RG
- Saída:** $(A_2, X, e_{K+1}, \dots, e_{\bar{K}}, CPG, RG)$

LRC formados por padrões de corte: Para a heurística de construção da solução inicial na qual a LRC é composta de padrões de corte, a LRC é construída sempre que um padrão de corte é inserido na solução inicial e é composta pelos índices 1, 2, 3, 4 e 5. Cada índice da LRC está relacionado a uma maneira de gerar um padrão de corte. Para gerar um padrão de corte, um índice é escolhido aleatoriamente na LRC. Esta heurística, denominada CSI_3 , está apresentada no Procedimento 6.

Procedimento 6 CSI_3

Entrada: $(m, K, \bar{K}, \ell_1, \dots, \ell_m, d_1, \dots, d_m, L_{K+1}, \dots, L_{\bar{K}}, e_{K+1}, \dots, e_{\bar{K}}, X, \alpha)$

1 **Enquanto** a demanda não foi atendida, **faça**

2 **Para** $j = K + 1, \dots, n$, **faça**

3 Selecione aleatoriamente um objeto L_k , para $k = K + 1, \dots, \bar{K}$

4 Crie a lista restrita de candidatos: $LRC = \{1, 2, 3, 4, 5\}$

5 Selecione aleatoriamente um elemento q de LRC e obtenha o padrão de corte a_{jk} da seguinte forma:

6 **Se** $q = 1$, **então**

7 | determine um padrão de corte resolvendo um problema da mochila restrito

8 **Fim**

9 **Se** $q = 2$, **então**

10 | insira itens aleatoriamente no padrão de corte a_{jk}

11 **Fim**

12 **Se** $q = 3$, **então**

13 | insira itens em ordem decrescente de tamanho no padrão de corte a_{jk}

14 **Fim**

15 **Se** $q = 4$, **então**

16 | insira itens no padrão de corte a_{jk} na ordem em que estão listados

17 **Fim**

18 **Se** $q = 5$, **então**

19 | insira itens em ordem de maior demanda no padrão de corte a_{jk}

20 **Fim**

21 Calcule: $x_{jk} = \min \left\{ e_k, \min_{1 \leq i \leq m} \left\lfloor \frac{d_i}{a_{ijk}} \right\rfloor \text{ tal que } a_{ijk} \neq 0 \right\}$

22 Atualize: $d_1, \dots, d_m, e_{K+1}, \dots, e_{\bar{K}}, A_2$ e X

23 **Fim**

24 **Fim**

25 Calcule: CPG, RG

Saída: $(A_2, X, e_{K+1}, \dots, e_{\bar{K}}, CPG, RG)$

3.3.2 Fase de melhoria da solução inicial

Para a fase de melhoria da solução inicial, começamos removendo da solução todos os padrões de corte que produzem perdas inaceitáveis (perdas no intervalo (ε, δ)). Depois disso, analisamos os padrões de corte restantes e se possível, modificamos para melhorar a solução inicial. Em seguida, adotamos dois critérios para melhorar os padrões de corte restantes, o que caracteriza as duas heurísticas usadas para a fase de melhoria da solução, denominadas Melhoria₁ e Melhoria₂.

O Procedimento 7, apresentado a seguir, descreve uma das formas para a fase de melhoria da solução inicial.

Procedimento 7 Melhoria₁

Entrada: $(m, K, \bar{K}, \ell_1, \dots, \ell_m, d_1, \dots, d_m, e_{K+1}, \dots, e_{\bar{K}}, L_{K+1}, \dots, L_{\bar{K}}, A_2, X, \varepsilon, \delta, p_{jk}$ (para todo j e k))

- 1 Exclua todos os padrões de corte com perda não aceitável da solução inicial
 - 2 Calcule a quantidade de padrões de corte produzindo retalho (k_1)
 - 3 Calcule a quantidade de padrões de corte produzindo perda (k_2)
 - 4 **Para** $it = 1, \dots, k_1$, **faça**
 - 5 Selecione o padrão de corte a_{jk} que produz a maior sobra p_{jk}
 - 6 Determine a^M resolvendo um problema da mochila restrito com capacidade p_{jk} e o acrescente ao padrão de corte a_{jk}
 - 7 Calcule quantas vezes o padrão de corte a_{jk} pode ser utilizado
 - 8 Atualize: $d_1, \dots, d_m, e_{K+1}, \dots, e_{\bar{K}}, A_2$ e X
 - 9 **Fim**
 - 10 **Para** $it = 1, \dots, k_2$, **faça**
 - 11 Selecione o padrão de corte a_{jk} que produz a maior sobra p_{jk} contando sua frequência
 - 12 Retire em ordem crescente de tamanho a metade da quantidade total de itens inseridos no padrão de corte a_{jk} e atualize p_{jk} e demanda de itens
 - 13 Determine a^M resolvendo um problema da mochila restrito com capacidade p_{jk} e o acrescente ao padrão de corte a_{jk}
 - 14 Calcule quantas vezes o padrão de corte a_{jk} pode ser utilizado
 - 15 Atualize: $d_1, \dots, d_m, e_{K+1}, \dots, e_{\bar{K}}, A_2$ e X
 - 16 **Fim**
 - 17 **Enquanto** a demanda não é atendida, **faça**
 - 18 Determine o padrão de corte a_{jk} resolvendo um problema da mochila restrito com capacidade L_k , em que L_k é selecionado aleatoriamente do estoque
 - 19 **Enquanto** a sobra do padrão de corte a_{jk} pertencer ao intervalo $(\frac{\varepsilon}{4}, \delta)$, **faça**
 - 20 Retire o item de menor comprimento do padrão de corte.
 - 21 **Fim**
 - 22 Calcule quantas vezes o padrão de corte a_{jk} pode ser utilizado
 - 23 Atualize: $d_1, \dots, d_m, e_{K+1}, \dots, e_{\bar{K}}, A_2$ e X
 - 24 **Fim**
 - 25 Atualize: CPG, RG
- Saída:** $(A_2, X, e_{K+1}, \dots, e_{\bar{K}}, CPG, RG)$
-

O Procedimento 8, apresentado a seguir, descreve uma das formas para a fase de melhoria da solução inicial.

Procedimento 8 Melhoria₂**Entrada:** $(m, K, \bar{K}, \ell_1, \dots, \ell_m, d_1, \dots, d_m, e_{K+1}, \dots, e_{\bar{K}}, L_{K+1}, \dots, L_{\bar{K}}, A_2, X, \varepsilon, \delta, p_{jk}$ (para todo j e k))

- 1 Exclua todos os padrões de corte com perda não aceitável da solução inicial
 - 2 Calcule a quantidade de padrões de corte produzindo sobra (k_1)
 - 3 **Para** $it = 1, \dots, k_1$, **faça**
 - 4 Selecione o padrão de corte a_{jk} que produz a maior sobra p_{jk} contando sua frequência
 - 5 Retire em ordem crescente de tamanho a metade da quantidade total de itens inseridos no padrão de corte a_{jk} e atualize p_{jk} e demanda de itens
 - 6 Determine o padrão de corte a^M resolvendo um problema da mochila restrito com capacidade p_{jk} e o acrescente ao padrão de corte a_{jk}
 - 7 **Enquanto** a sobra do padrão de corte a_{jk} pertencer ao intervalo $\left(\frac{\delta}{2}, \delta\right)$, **faça**
 - 8 Retire o item de menor comprimento do padrão de corte.
 - 9 **Fim**
 - 10 Calcule quantas vezes o padrão de corte a_{jk} pode ser utilizado
 - 11 Atualize: $d_1, \dots, d_m, e_{K+1}, \dots, e_{\bar{K}}, A_2$ e X
 - 12 **Fim**
 - 13 **Enquanto** a demanda não é atendida, **faça**
 - 14 Determine o padrão de corte a_{jk} resolvendo um problema da mochila restrito com capacidade L_k , em que L_k é selecionado aleatoriamente do estoque
 - 15 **Enquanto** a sobra do padrão de corte a_{jk} pertencer ao intervalo $\left(\frac{\delta}{2}, \delta\right)$, **faça**
 - 16 Retire o item de menor comprimento do padrão de corte
 - 17 **Fim**
 - 18 Calcule quantas vezes o padrão de corte a_{jk} pode ser utilizado
 - 19 Atualize: $d_1, \dots, d_m, e_{K+1}, \dots, e_{\bar{K}}, A_2$ e X
 - 20 **Fim**
 - 21 Atualize: CPG, RG
- Saída:** $(A_2, X, e_{K+1}, \dots, e_{\bar{K}}, CPG, RG)$

3.3.3 Atualizar solução

Depois de encontrar uma solução inicial e submetê-la ao procedimento de melhoria da solução, analisamos a solução atual armazenada e a solução da iteração para registrar a melhor solução considerando o seguinte critério:

Registrar: A melhor solução é aquela com a menor soma do comprimento total de perdas (PGC) com a quantidade de retalhos gerados (RG). Se, em uma determinada iteração, a soma do comprimento total de perdas com a quantidade de retalho gerado for igual ao que é armazenado até então, a quantidade total de retalho gerado é analisada e armazenada como a melhor solução com a menor quantidade;

Com as três heurísticas projetadas para construir uma solução inicial, as duas heurísticas para melhorar a solução inicial e a maneira de armazenar a melhor solução, foram desenvolvidas três metaheurísticas GRASP, chamadas GRASP₁, GRASP₂ e GRASP₃. As metaheurísticas propostas são apresentadas em um esquema na Figura 3.

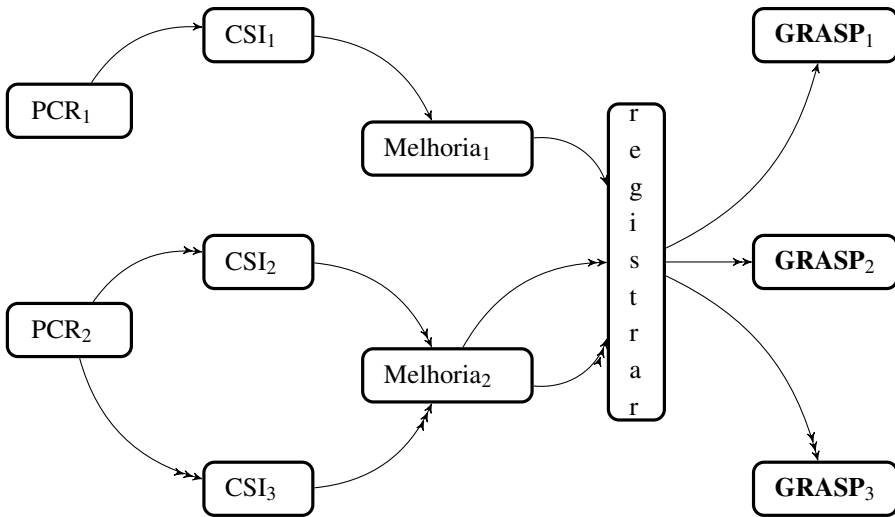


Figura 3: Esquema para as diferentes formas de obter uma solução utilizando as heurísticas PCR₁ e PCR₂, juntamente com as metaheurísticas GRASP₁, GRASP₂ e GRASP₃.

4 TESTES COMPUTACIONAIS

Vários testes computacionais foram realizados para verificar o desempenho dos procedimentos propostos usando o *software* MATLAB versão 7.10.0.499(R2010a) e CPLEX versão 12.6.1 em um microcomputador processador Intel (R) Core (TM) i7 – 4790 CPU 3.60 GHz, com memória RAM de 16.00 Gb, sistema operacional Windows 10. Os resultados computacionais foram comparados com os obtidos por Nicola [12] para as mesmas instâncias usando o *software* DELPHI 6 sendo executado em um microcomputador Pentium IV (3 GHz, 2 GB RAM). Dezesesseis classes distintas foram usadas, com vinte instâncias cada, totalizando 320 instâncias.

A Tabela 1 mostra as características de cada classe, onde o comprimento de cada item foi gerado aleatoriamente no intervalo $[v_1L, v_2L]$, com $v_1 = 0,01$ e L sendo o comprimento médio dos objetos padronizados utilizados (Foram considerados dois tipos de objetos padronizados). Para v_2 , temos que:

- (a) $v_2 = 0,2$, as classes são compostas por itens com pouca heterogeneidade: itens do tipo P;
- (b) $v_2 = 0,8$, as classes são compostas por itens com muita heterogeneidade: itens do tipo M.

O Comprimento da Perda Gerada (CPG) é qualquer perda gerada no plano de corte, que inclui padrões de corte construídos com objetos padronizados e não padronizados. O Retalho Gerado (RG) é a quantidade de retalhos gerados na obtenção do plano de corte, que inclui apenas padrões de corte gerados com os objetos padrão, já que os padrões de corte gerados com os objetos não padronizados não produzem retalhos. O Retalho Usado (RU) é a quantidade de objetos não padronizados utilizados do estoque. O Retalho Final (RF) é a quantidade de retalhos no estoque

Tabela 1: Descrição das classes.

Classe	\bar{K}	m	Heterogeneidade
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

final, ou seja, a quantidade de objetos não padronizados no estoque subtraindo RU e adicionando RG. Os parâmetros considerados na metaheurística GRASP foram definidos na Seção 3.

A Tabela 2 ilustra os resultados obtidos em Nicola [12] e a Tabela 3 ilustra os resultados obtidos pelas heurísticas PCR₁, PCR₂ e pelas metaheurísticas GRASP₁, GRASP₂ e GRASP₃.

Nas colunas RF das Tabela 2 e Tabela 3, observamos que, em 87,5% das classes na metaheurística GRASP₁ e 81,25% das classes nas Metaheurísticas GRASP₂ e GRASP₃, ao encontrar uma solução para os problemas testados, reduziu o estoque de retalhos. No entanto, houve um aumento no número de retalho no estoque final em duas classes comparadas nas metaheurísticas GRASP₁ e três classes comparadas nas metaheurísticas GRASP₂ e GRASP₃.

O estoque final foi o melhor resultado encontrado com o uso dos procedimentos propostos. Nesse caso, quando consideramos o melhor valor encontrado em Nicola [12] e o pior resultado encontrado neste artigo, ainda temos um resultado melhor em 42% dos retalhos gerados.

Na análise dos resultados, observamos que houve uma diferença no desempenho do método em relação ao comprimento da perda e quantidade de retalhos gerados para os dois tipos de itens considerados nas instâncias, pouca e muita heterogeneidade. Assim, para esses dois casos, perdas e retalhos gerados, fizemos uma análise separadamente para os tipos de itens considerados P e M.

Para as classes compostas de itens com pouca heterogeneidade, considerando a média de todas as classes com itens do tipo P, pelas colunas referentes à CPG da Tabela 2 e Tabela 3, podemos

Tabela 2: Resultados computacionais referentes ao comprimento das perdas geradas (CPG), quantidade de retalhos gerados (RG), quantidade de objetos não padronizados utilizados (RU) e quantidade de retalho no estoque final (RF) para Gulosa_A, RAG_{A1}, RAG_{A2} e RAG_{A3}, heurísticas propostas em Nicola [12].

	Gulosa _A				RAG _{A1}				RAG _{A2}				RAG _{A3}			
	CPG	RG	RU	RF	CPG	RG	RU	RF	CPG	RG	RU	RF	CPG	RG	RU	RF
C ₁	15,00	2,70	8,60	-5,90	7,80	2,20	4,80	-2,60	7,10	2,30	4,90	-2,60	6,60	2,00	4,50	-2,50
C ₂	40,20	4,20	4,50	-0,30	87,60	11,30	4,20	+7,10	97,30	9,30	4,40	+4,90	93,80	10,80	4,30	+6,50
C ₃	12,60	2,10	8,70	-6,60	3,20	2,20	4,50	-2,30	3,30	2,20	4,50	-2,30	3,40	2,50	4,50	-2,00
C ₄	32,30	4,10	4,40	-0,30	18,80	4,90	4,10	+0,80	18,70	4,80	4,20	+0,60	24,40	4,70	3,50	+1,20
C ₅	8,50	1,60	5,50	-3,90	3,50	2,10	3,90	-1,80	3,50	2,20	3,90	-1,70	1,70	2,00	4,20	-2,20
C ₆	74,50	1,80	6,10	-4,30	7,40	1,60	3,40	-1,80	7,40	1,60	3,40	-1,80	6,70	2,10	3,00	-0,90
C ₇	11,00	3,90	11,70	-7,80	2,40	2,10	4,60	-2,50	2,40	2,10	4,60	-2,50	3,30	1,60	4,80	-3,20
C ₈	54,30	9,00	11,20	-2,20	19,40	3,10	6,20	-3,10	17,00	3,00	6,10	-3,10	21,40	3,30	5,40	-2,10
C ₉	11,10	2,60	9,90	-7,30	1,20	1,50	4,20	-2,70	1,20	1,50	4,20	-2,70	1,90	1,80	4,30	-2,50
C ₁₀	20,10	7,20	11,60	-4,40	8,10	1,50	4,10	-2,60	7,40	1,70	4,20	-1,50	8,80	1,60	3,50	-1,90
C ₁₁	11,70	0,60	8,70	-8,10	1,60	1,80	3,80	-2,00	1,70	1,70	3,70	-2,00	3,90	2,60	6,20	-3,90
C ₁₂	37,80	2,80	10,10	-7,30	4,00	1,10	2,60	-1,50	3,80	1,20	2,60	-1,40	2,80	1,70	3,80	-2,10
C ₁₃	16,30	3,20	12,00	-8,80	1,10	1,70	4,40	-2,70	1,00	1,70	4,40	-2,70	1,10	1,50	3,60	-2,10
C ₁₄	49,60	1,30	13,50	-12,20	11,50	1,80	5,60	-3,80	12,00	2,10	6,00	-3,90	12,50	2,80	6,80	-4,00
C ₁₅	20,90	3,00	13,50	-10,50	1,40	1,30	4,00	-2,70	1,30	1,40	4,20	-2,80	2,00	2,20	5,00	-2,80
C ₁₆	50,60	4,10	14,30	-10,20	4,30	2,10	4,90	-2,80	4,30	2,10	4,90	-2,80	5,60	1,50	3,80	-2,30

Tabela 3: Resultados computacionais referentes ao comprimento das perdas geradas (CPG), quantidade de retalhos gerados (RG), quantidade de objetos não padronizados do estoque utilizados (RU) e quantidade de retalho no estoque final (RF) para as heurísticas PCR₁ e PCR₂ e metaheurísticas GRASP₁, GRASP₂ e GRASP₃ propostas neste artigo.

	GRASP ₁				GRASP ₂				GRASP ₃			
	CPG	RG	RU	RF	CPG	RG	RU	RF	CPG	RG	RU	RF
C ₁	0,30	1,05	16,65	-15,60	0,95	1,00	16,65	-15,65	0,40	1,00	16,65	-15,65
C ₂	190,25	29,50	6,25	+23,25	75,20	33,75	7,70	+30,00	88,55	32,95	7,70	+25,25
C ₃	0,65	1,20	16,30	-15,10	0,30	1,20	16,30	-15,10	0,00	1,05	16,30	-15,25
C ₄	95,35	15,60	12,40	+3,20	48,15	21,00	11,45	+9,55	56,65	20,05	11,45	+18,60
C ₅	0,00	0,00	14,70	-14,70	0,10	0,15	14,70	-14,55	0,05	0,65	14,70	-14,05
C ₆	17,75	1,40	9,25	-7,85	47,50	11,95	9,25	+2,70	47,30	9,40	9,25	+0,15
C ₇	0,20	1,00	27,20	-26,20	0,65	1,00	27,30	-26,30	0,35	1,00	27,30	-26,30
C ₈	45,80	10,00	16,40	-6,40	31,70	11,80	17,75	-5,95	34,80	11,70	17,75	-6,05
C ₉	0,00	1,00	28,10	-27,10	0,20	1,10	28,10	-27,00	0,00	1,00	28,10	-27,10
C ₁₀	13,30	12,70	24,05	-11,35	9,75	8,05	24,15	-16,10	11,20	7,85	24,15	-14,30
C ₁₁	0,00	0,00	25,50	-25,50	0,10	0,45	25,50	-25,05	0,00	0,95	25,50	-24,55
C ₁₂	0,40	0,05	18,85	-18,80	11,50	2,35	18,85	-16,50	9,35	2,70	18,85	-16,15
C ₁₃	0,25	1,00	40,85	-39,85	0,25	1,75	40,85	-39,10	0,15	1,00	40,85	-39,85
C ₁₄	14,05	6,55	30,15	-23,60	20,35	8,65	32,10	-23,45	25,55	8,25	32,10	-23,85
C ₁₅	0,00	1,00	40,00	-39,00	0,00	1,00	40,00	-39,00	0,00	1,00	40,00	-39,00
C ₁₆	8,30	5,85	34,25	-28,40	8,00	6,50	33,65	-27,15	5,80	7,50	33,65	-26,15

afirmar que: usando a metaheurística GRASP₁ obtemos uma média do comprimento de perdas de 0,17, usando a metaheurística GRASP₂ obtemos uma média do comprimento das perdas de 0,31 e com o metaheurística GRASP₃ obtemos uma média do comprimento das perdas de 0,11. A metaheurística GRASP₃ obteve o melhor resultado; nesse caso, o menor valor encontrado por Nicola [12] é cerca de 22 vezes maior. Podemos dizer que, nas classes com itens com pouca heterogeneidade, em média, as perdas resultantes da solução foram menores nas metaheurísticas propostas.

No entanto, nas classes compostas por itens muito heterogêneos, não podemos dizer que as metaheurísticas propostas tiveram um bom desempenho em comparação às heurísticas comparadas.

Pelas colunas referentes à RG da Tabela 2 e da Tabela 3, podemos afirmar que: nas classes com itens pouco heterogêneos, todas as metaheurísticas tiveram bons resultados para a quantidade de retalhos gerados na solução; nas classes com itens muito heterogêneos, quando há poucos itens, também a quantidade de retalhos gerados não é satisfatória. Para as classes compostas de itens do tipo P, em relação à quantidade de retalho gerado, a metaheurística GRASP₁ obteve o melhor resultado; nesse caso, o menor valor encontrado em Nicola [12] é cerca de duas vezes maior.

Em relação aos retalhos gerados, obtemos que 62,5% das classes na metaheurística GRASP₁, 43,75% na metaheurística GRASP₂ e 50% na metaheurística GRASP₃ geraram a menor quantidade de todas as heurísticas comparadas.

A partir das colunas referentes à RU da Tabela 2 e Tabela 3, observamos que a quantidade de objetos não padronizados utilizados na solução com as abordagens propostas foi maior que nas heurísticas comparadas. Foram utilizados 81% com o uso de PCR₁ e 81,14% com o uso de PCR₂ dos objetos não padronizados na obtenção do plano de corte.

Para classes compostas por itens muito heterogêneos, não foram encontrados bons resultados. Isso ocorre devido ao fato de que os padrões de corte gerados nas classes compostas por itens demandados com pouca heterogeneidade são construídos com média de 31,24 itens em cada objeto, e os padrões de corte gerados nas classes compostas por itens com muita heterogeneidade são construídos com, em média, 8,03 itens em cada objeto, dificultando uma boa combinação para gerar um bom padrão de corte.

Com relação ao tempo computacional, observamos que, para resolver os Procedimentos PCR₁ ou PCR₂, em conjunto com a metaheurística GRASP₁ ou GRASP₂ ou GRASP₃, foi necessário um tempo maior que o tempo usado para resolver as heurísticas comparadas. De acordo com Nicola [12], os 320 exemplos foram resolvidos com um tempo médio de 0,367s, 22,55s, 22,7s e 81,85s, respectivamente, nas heurísticas Gulosa_A, RAG_{A1}, RAG_{A2} e RAG_{A3}. Com o uso das metaheurísticas GRASP₁, GRASP₂ e GRASP₃, obtivemos um tempo médio de 141,11s, 117,97s e 130,74s, respectivamente, na resolução das instâncias.

5 CONCLUSÃO

Neste trabalho, realizamos um estudo sobre o problema de corte de estoque unidimensional com sobras aproveitáveis. Três versões da meta-heurística GRASP, as quais foram chamadas de GRASP₁, GRASP₂ e GRASP₃, e dois procedimentos para gerar padrões de corte a partir de objetos não padronizados, chamados de PCR₁ e PCR₂, foram desenvolvidas, implementadas e realizados testes computacionais com dados fornecidos por Nicola [12]. Para isso, foram utilizados os *softwares* MATLAB e CPLEX. Os resultados encontrados com o uso das heurísticas e metaheurísticas propostas foram comparados com os resultados apresentados em [12] com o uso das heurísticas GulosaA, RAG_A1, RAG_A2 e RAG_A3 utilizando os mesmos exemplares.

Na análise dos resultados obtidos pelas metaheurísticas propostas, podemos afirmar que, quando analisada a média de todas as classes testadas, a metaheurística GRASP₃ obteve o melhor resultado, pois além de gerar a menor perda para as classes com itens do tipo P obteve uma pequena quantidade de retalhos gerados e uma grande redução no número de retalhos no estoque final. Para as classes compostas por itens do tipo M, o melhor resultado médio dessas classes para as perdas geradas foi encontrado usando a metaheurística GRASP₂. Em relação à quantidade de retalhos gerados na solução, podemos afirmar que a metaheurística GRASP₁ obteve os melhores resultados quando analisada a média de todas as classes e também nos casos de classes com itens do tipo P e M.

Para a quantidade de objetos padronizados e não padronizados utilizados, bem como para o estoque final de retalhos, analisamos a comparação de todas as classes e obtivemos o melhor desempenho na metaheurística GRASP₁. Nesses três casos, a metaheurística GRASP₁, quando comparada às heurísticas propostas em Nicola [12], também teve um desempenho melhor. Para os procedimentos propostos neste artigo, o PCESA foi resolvido em tempo computacional razoável.

AGRADECIMENTOS

Esta pesquisa foi financiada pela CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

ABSTRACT. In this article we present a study of the one-dimensional Cutting Stock Problem with Usable Leftovers Stock (CSPUL). The Cutting Stock Problem (CSP) consists of cutting a set of objects, available in stock, to produce a set of items in specified quantities and sizes. CSPUL is a CSP in which not all leftovers from a cutting process are considered waste, that is, when a leftover is greater than a certain length, it returns to the stock to be used in the subsequent cutting processes. Therefore, the objective of the addressed problem is to optimize the cutting planes to be applied to standardized objects and non-standardized objects (leftovers from previous cuts); meet the known demand for smaller items, in order to minimize the waste of material, controlling the amount of generated leftovers. In order to solve this problem, two heuristics and three versions of the *Greedy Randomized Adaptive Search Procedure* metaheuristic (GRASP) were developed. Computational tests were per-

formed for the procedures proposed in instances of the literature. The approaches proposed for the solution presented good solutions in relation to the number of surpluses generated.

Keywords: one-dimensional cutting stock problem, usable leftovers, metaheuristics, GRASP.

REFERÊNCIAS

- [1] A. Abuabara & R. Morabito. Cutting optimization of structural tubes to build agricultural light aircrafts. *Annals of Operations Research*, **169**(1) (2009), 149–165. URL <https://doi.org/10.1007/s10479-008-0438-7>.
- [2] M.N. Arenales, A.C. Cherri, D.N. Nascimento & A. Vianna. A new mathematical model for the cutting stock/leftover problem. *Pesquisa Operacional*, **35**(3) (2015), 509–522.
- [3] A.R. Brown. Optimum packing and depletion: the computer in space and resource usage problem. *Macdonald and Co.*, (1971).
- [4] A.C. Cherri, M.N. Arenales & H.H. Yanasse. The one-dimensional cutting stock problem with usable leftover—A heuristic approach. *European Journal of Operational Research*, **196**(3) (2009), 897–908.
- [5] A.C. Cherri, M.N. Arenales & H.H. Yanasse. The usable leftover one-dimensional cutting stock problem—a priority-in-use heuristic. *International Transactions in Operational Research*, **20**(2) (2013), 189–199.
- [6] A.C. Cherri, M.N. Arenales, H.H. Yanasse, K.C. Poldi & A.C.G. Vianna. The one-dimensional cutting stock problem with usable leftovers—A survey. *European Journal of Operational Research*, **236**(2) (2014), 395–402.
- [7] T.A. Feo & M.G. Resende. Greedy randomized adaptive search procedures. *Journal of global optimization*, **6**(2) (1995), 109–133.
- [8] M.R. Garey & D.S. Johnson. “Computers and intractability: a guide to the theory of NP-completeness”. WH Free. Co., São Francisco (1979).
- [9] M. Gradišar, J. Jesenko & G. Resinovič. Optimization of roll cutting in clothing industry. *Computers & Operations Research*, **24**(10) (1997), 945–953.
- [10] S. Koch, S. König & G. Wäscher. Integer linear programming for a cutting problem in the wood-processing industry: a case study. *International Transactions in Operational Research*, **16**(6) (2009), 715–726.
- [11] E.V. Neto, A.S. Velasco & G.G. de Paula Junior. ALGORITMOS GRASP HÍBRIDOS PARA O PROBLEMA DE CORTE UNIDIMENSIONAL. *XLV Simpósio Brasileiro de Pesquisa Operacional*, (2013).
- [12] A.C.C. Nicola. “Algumas extensões do problema de corte de estoque com sobras de material aproveitáveis”. Tese de doutorado, ICMC - USP, São Carlos (2009).
- [13] M.G. Resende & J.P. de Sousa. “Metaheuristics: computer decision-making”, volume 86. Springer Science & Business Media (2013).

- [14] G.M. Roodman. Near-optimal solutions to one-dimensional cutting stock problems. *Computers & operations research*, **13**(6) (1986), 713–719.
- [15] G. Scheithauer. A note on handling residual lengths. *Optimization*, **22**(3) (1991), 461–466.