GERAÇÃO COLUNAS E GRASP REATIVO PARA O PROBLEMA DE CORTE UNIDIMENSIONAL

João Vitor Bernardo Moreira da Silva
Instituto Federal Fluminense - IFF
Rua Dr. Siqueira, 273. Parque Dom Bosco - Campos dos Goytacazes - RJ. CEP: 28030-130
joaoavitorbmds@gmail.com

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

RESUMO

Este trabalho aborda o Problema de Corte Unidimensional e propõe dois algoritmos, sendo um heurístico e o outro híbrido, para sua resolução. Utilizando os principais elementos do algoritmo híbrido GRASP-1D em sua implementação, o algoritmo GR_{1d} se diferencia com uma abordagem reativa que identifica inicialmente o parâmetro de aleatoriedade α que melhor se adequa a instância corrente. Já o algoritmo híbrido GCH_{1d} caracteriza-se como híbrido porque é inicializado com a técnica Geração de Colunas, onde subproblema é um problema de programação inteira, resolvido de forma exata para retornar o padrão de corte maximal. Se a solução ótima do PL Mestre não é inteira, determina-se a demanda do problema residual que deve ser resolvido pela heurística GR_{1d}. Com o objetivo de avaliar o desempenho destes algoritmos, testes são executados instâncias reais. Ao confrontar os resultados com algoritmos híbridos GRASP-1D, ALG HB1 e ALG HB2, contatou-se um desempenho significativo do algoritmo GCH_{1d}.

PALAVRAS CHAVE. Padrões de Corte Unidimensionais, Geração de Colunas, GRASP Reativo.

Tópicos (indique, em ordem de PRIORIDADE, o(s) tópicos(s) de seu artigo): MH, OC

ABSTRACT

This work approaches the One-Dimensional Cutting Problem and proposes two algorithms for its resolution, one heuristic and the other hybrid. Using the main elements of the GRASP-1D hybrid algorithm in its implementation, GR_{1d} algorithm differs with a reactive approach that initially identifies the randomness parameter α that best fits the current instance. The hybrid algorithm GCH_{1d} is characterized as hybrid because it is initialized with the Column Generation technique, where a sub-problem is an integer programming problem, solved in an exact way to return the maximal cut pattern. If the optimal solution of the Master Linear Problem is not integer, the demand of the residual problem that must be solved by the GR_{1d} heuristic is determined. in order to evaluate their performances, tests are run on real instances. When comparing the results with GRASP-1D, ALG HB1 and ALG HB2 hybrid algorithms, a significant performance of the GCH_{1d} algorithm was found.

KEYWORDS. One-dimensional Cutting Patterns. Column Generation. Reactive GRASP.

Paper topics (indicate in order of PRIORITY the paper topic(s)): MH, OC

1. Introdução

Problema de Corte de Estoque Unidimensional clássico, abreviado neste trabalho como Problema de Corte Unidimensional (PCU), consiste em determinar como se deve cortar um conjunto de objetos, cuja a única dimensão relevante é seu comprimento, com quantidade suficiente para gerar itens de menor comprimento e com demandas pré-estabelecidas, utilizando a menor quantidade possível desses objetos.

A importância deste problema na prática se torna evidente em vista do grande interesse no planejamento da produção de alguns setores industriais, que consideram ações de corte na transformação de materiais em produtos semiacabados ou finais, como por exemplo: construção civil, metal mecânica, cartonagem, entre outros.

Com o PCU pertencente a classe NP-difícil [Garey e Johnson 1979], a utilização de algoritmos exatos demanda de esforço computacional considerável, inviabilizando a sua utilização em situações práticas. Entre as abordagens desenvolvidas na literatura para resolver o problema em destaque, a clássica técnica Geração de Colunas [Gilmore e Gomory 1961] e a metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP) [Vieira Neto et al. 2010] são uma alternativa válida, evidenciadas inclusive em outras variantes do PCU [Rios e Poldi 2020]. Objetivando gerar boas soluções, possivelmente até a solução ótima, em tempos de execução que estejam de acordo com as necessidades cotidianas das empresas, são propostos os algoritmos GR_{1d} e GCH_{1D} para resolução do PCU, sendo o primeiro uma heurística que utiliza os conceitos do método GRASP Reativo [Prais e Ribeiro 2000] para propor uma adaptação no algoritmo GRASP-1D apresentado em [Vieira Neto et al. 2013], e o segundo constituindo-se como uma composição híbrida entre a Geração de Colunas e o GR_{1d}, baseado no trabalho de [Velasco e Uchoa 2015] para o Problema de Corte de Estoque Bidimensional (PCEB).

Assim exposto, verifica-se na literatura a utilização heurísticas ou algoritmos híbridos para resolução do PCU e suas variantes, com algumas destas abordagens sendo corroboradas nos seguintes trabalhos: [Gilmore e Gomory 1961]; [Gilmore e Gomory 1963]; [Stadtler 1990]; [Cintra 1998]; [Foerster e Wäscher 2000]; [Vieira Neto 2004]; [Yanasse e Limeira 2006]; [Cui et al. 2008]; [Vieira Neto et al. 2013]; [Rios e Poldi 2020].

Para avaliar o desempenho dos algoritmos desenvolvidos GR_{1d} e GCH_{1d}, estes são executados em um grupo de 35 instâncias práticas, retiradas de fábricas e disponibilizadas em [Vieira Neto 2004]. Tais soluções são confrontadas com as geradas pelos algoritmos híbridos denominados GRASP-1D, ALG HB1 e ALG HB2 [Vieira Neto 2013].

O artigo está dividido em 6 seções. Uma breve apresentação da formulação do PCU e os conceitos fundamentais do método Geração de Colunas serão apresentadas na Seção 2. A Seção 3 apresenta a metodologia GRASP e sua abordagem reativa na implementação da heurística GR_{1d}. Na Seção 4, será descrito o algoritmo híbrido GCH_{1d}, desenvolvido numa ação conjunta das técnicas Geração de Colunas e o algoritmo GRASP Reativo apresentado na seção anterior. Na Seção 5, serão mostrados os testes computacionais bem como a análise dos resultados. A Seção 6 será destinada às conclusões e trabalhos futuros.

2. Formulação do PCU

Admitindo que são conhecidos os padrões de corte unidimensionais, que garantam atender uma demanda de itens a partir de objetos em estoque em quantidade suficiente, o PCU pode ser descrito como um problema de Programação Linear Inteira. Na formulação apresentada a seguir, tem-se o objetivo de determinar o número de vezes que cada padrão de corte é utilizado, de forma a atender tal demanda, utilizando o menor número possível de objetos.

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

$$\text{s.a:} \quad \sum_{j=1}^{n} a_{ij}\, x_j \geq d_i \,, \quad i = 1,...,m \qquad (2.1)$$

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

Onde:
- x_j é a variável atribuída ao número de vezes que o padrão de corte j é utilizado;
- a_{ij} é a quantidade de itens i produzidos no padrão de corte j;
- d_i é a quantidade demandada do item i;
- m é a quantidade total de itens distintos;
- n é a quantidade total de padrões de corte.

A condição de integridade imposta as variáveis x_j e a desvantagem computacional de se produzir todos os padrões de corte, considerando aumento do número m de itens, impede que a Relaxação Linear dessa formulação seja resolvida diretamente por um método como o Simplex. Esta dificuldade pode ser contornada com o método de geração de colunas [Dantizg e Wolfe 1960], que também é considerado como uma extensão do Simplex Revisado, onde o *pricing* das variáveis é realizado através da resolução de um subproblema. A Geração de Colunas para o PCU foi introduzida nos trabalhos de [Gilmore e Gomory 1961] e [Gilmore e Gomory 1963], onde o chamado Problema Linear Mestre (PL Mestre) é inicializado com apenas um pequeno subconjunto das variáveis, correspondendo a alguns padrões de corte. A cada iteração, é gerado e inserido um novo padrão, desde que ele possa melhorar o valor da função objetivo na formulação supracitada (2.1), até que se prove que a solução corrente é ótima. Sendo um problema de minimização, deve-se buscar por padrões que levem a variáveis com custo reduzido negativo. Isso resulta no seguinte subproblema:

$$\begin{aligned} \text{Min} \quad & (1 - \sum_{i=1}^{m} \pi_i y_i) \\ \text{s.a:} \quad & [y_1,..., y_m]^t \text{ é um padrão viável} \\ & y_i \geq 0 \text{ e Inteiro} \end{aligned} \qquad (2.2)$$

Onde:
- y_i é a variável atribuída ao número de itens i no padrão de corte de tamanho L;
- π_i são as variáveis duais para cada restrição i do PL Mestre.

Neste presente trabalho, o subproblema (2.2) é tratado como problema da mochila inteiro, que resulta em gerar um padrão de corte unidimensional com máximo valor de utilidade nas respectivas variáveis duais.

A relaxação contínua do modelo (2.1) é considerada muito forte, no sentido em que o limite inferior obtido costuma ser muito próximo da solução ótima [Marcote 1986]. A conjectura *Modified Integer Round-Up Property* (Conjectura MIRUP), proposta por [Scheithauer e Terno 1995], assinala que o limite inferior do modelo (2.1) arredondado para cima está no máximo uma unidade abaixo do valor da solução ótima inteira. No momento presente, a maior diferença conhecida entre o limite inferior e a solução ótima é de 1,16666... [Rietz et al. 2002]. Entretanto, apesar de um grande esforço de pesquisa, ainda não foi possível provar essa conjectura.

3. Algoritmo GR_{1d}

A metaheurística GRASP foi proposta por [Feo e Resende 1989] e constitui-se de uma heurística construtiva associada a um procedimento de busca local. Com o algoritmo GR_{1d} sendo implementado para o PCU, inspirando-se no algoritmo GRASP-1D [Vieira Neto et al. 2013] e no estudo sobre a influência do parâmetro alfa na qualidade das soluções construídas [Vieira Neto et al. 2010]. As duas fases da GRASP, a primeira de construção e a segunda de melhoria, são descritas separadamente dando ênfase aos principais procedimentos apresentados nas referências em destaque e as especificidades frente ao presente trabalho.

3.1 Fase de Construção da GRASP para o PCU

Na primeira fase, cada padrão de corte é gerado iterativamente escolhendo um item de cada vez, selecionado de um conjunto restrito de itens denominado de lista restrita de candidatos (LRC). A LRC é composta pelos itens candidatos a serem incluídos no padrão corrente, seguindo um critério guloso que o atribui um valor de utilidade. Um parâmetro α determina a quantidade de itens incluídos no LRC, assumindo valores no intervalo [0,1]. Seja P um conjunto finito formado pelos próximos itens p_i a serem incluídos na solução, $f : P \rightarrow \mathbb{R}$ uma função gulosa que retorna o valor de utilidade do item (comprimento) e $\beta = \max \{f(p_i); p_i \in P\}$ o melhor valor retornado pela função gulosa, com i = 1,...,m. A LRC = $\{p_i \in P / \alpha.\beta \leq f(p_i) \leq \beta\}$ é determinada multiplicando o parâmetro α ao valor β para definição de um intervalo de valores. Assim definida, quando $\alpha = 1$ se gera soluções totalmente gulosas e para $\alpha = 0$ se produz soluções totalmente aleatórias. A cada rodada do algoritmo, a presente fase é concluída com uma solução do PCU, constituída de padrões de corte e suas respectivas repetições, que será direcionada a próxima fase.

3.2 Fase de Busca Local da GRASP para o PCU

A segunda fase é um procedimento que determina uma estrutura de vizinhança para gerar um subconjunto de soluções, a partir da solução construída na fase inicial, buscando escapar de ótimos locais, com o propósito de melhorá-la. Nesse contexto de vizinhança, buscando-se melhorar as soluções originadas na fase de construção, são escolhidos alguns padrões de corte, constituindo-se um PCU de menor tamanho (PCU_{BL}) que a instância original, na tentativa de reagrupar os itens que os compõem, objetivando minimizar o número de objetos utilizados para atender a demanda de tais itens. Os quatro critérios listados posteriormente, são adotados para selecionar os padrões de corte para compor uma instância PCU_{BL}.

(1)O padrão com a maior perda individual, ou seja, que tenha o maior comprimento da perda, considerando que o mesmo possa ser executado mais de uma vez;
(2)O padrão que gera a maior perda quando usado, ou seja, que o produto de sua perda individual pelo número de vezes que é executado resulta o maior valor;
(3)O padrão que contém o maior número de itens;
(4)O padrão que gera o maior número de itens quando usado, ou seja, o produto do número de itens do padrão pelo número de vezes que o mesmo é executado.

Diferentemente do híbrido GRASP-1D, onde os itens dos padrões retirados da solução inicial e que compõe o subproblema PCU_{BL} são submetidos a um gerador de padrões viáveis e resolvidos como um problema linear pelo método Simplex, relaxando a integridade de suas variáveis, a versão implementada para este trabalho (GR_{1d}) resolve tal subproblema de forma recursiva, utilizando os procedimentos supracitados para as duas fases da GRASP. Em ambas as abordagens, optou-se como critério de parada por um número máximo de iterações.

3.3 Algoritmo GRASP Reativo para o PCU

O método GRASP Reativo provém do GRASP clássico e, basicamente, diferencia-se na fase de construção, quando o parâmetro α é deliberado após uma fina calibragem feita a partir de uma variação deste parâmetro no decorrer de algumas iterações [Prais e Ribeiro 2000].

A implementação do algoritmo GR_{1d} apresenta duas etapas distintas, chamadas de fase de calibragem e fase reativa. Basicamente, no ajuste do parâmetro α realizado na fase de calibragem, inicialmente, considera-se um conjunto finito A de valores no intervalo [0, 1], pré-definidos de acordo com as características do problema. Com uma aprimorada calibragem após um determinado número de iterações $iter_C$, define-se o alfa reativo α_R como aquele que apresenta com maior frequência a melhor solução. É importante relatar, ainda, que em caso de empate das respectivas frequências, um sorteio estabelece o parâmetro que melhor se adequa à instância. Na fase posterior, denominada fase reativa, inicia-se com a determinação de um novo conjunto A' constituído de valores no entorno da mediana α_R, estabelecendo acréscimos e decréscimos a

serem realizados nesse parâmetro reativo. Assim como na fase inicial, executa-se nesta etapa as duas fases da GRASP para o PCU, com o total de $iter_R$ iterações para cada elemento de A', com o objetivo de escapar possíveis ótimos locais e produzir melhores soluções.

4. Algoritmo GCH_{1d}

O algoritmo híbrido GCH_{1d} foi implementado para o PCU, baseando-se na estratégia apresentada por [Gilmore e Gomory 1961], iniciando com a técnica de geração de colunas e o subproblema sendo resolvido por programação inteira e de forma exata, combinada com o algoritmo GR_{1d} para obter soluções inteiras aproximadas para o problema residual (PCU_R). Em sua rotina, o PL Mestre (2.1) começa com um conjunto de m colunas associadas a padrões homogêneos, com os itens produzidos em sua quantidade máxima em cada objeto. A cada iteração seguinte, se for vantajoso, atribui-se aos itens os valores das suas respectivas variáveis duais, gera e insere uma nova coluna ao PL Mestre. Supondo que a solução ótima do PL Mestre relaxado não seja inteira, ou seja, existe pelo menos um padrão de corte cujo número de repetições é um número decimal, consegue-se uma solução inteira aproximada arredondando para o maior inteiro menor ou igual à solução relaxada corrente. Nesse caso, determina-se um PCU_R, que geralmente possui tamanho menor que a instância original, desde que tal solução apresente alguma variável $x_j > 1$, para ser resolvido pelo algoritmo GR_{1d}, compondo assim a solução inteira do PCU original [Velasco e Uchoa 2015].

Na apresentação de um pseudocódigo para o algoritmo GCH_{1d}, destaca-se a definição das variáveis ainda não declaradas nos referentes procedimentos.

- S^*_{PCU} : solução do PCU;
- $valorS^*_{PCU}$: número total de objetos na solução do PCU;
- S^*_{Rel} : solução relaxada do PL Mestre;
- S_{Apr} : solução inteira aproximada do PL Mestre;
- $valorS_{Apr}$: número de objetos na solução inteira aproximada;
- $itens_{PR}$: número total de itens na instância residual.

Algoritmo GCH_{1d} (A, $iter_C$, $iter_R$, L, p_i,..., p_m, d_i,..., d_m)

1. $S^*_{PCU} = \varnothing$ e $valorS^*_{PCU} = 0$;
2. Determine as colunas $[y_1,..., y_m]^t$ de padrões homogêneos para o PL Mestre (2.1);
3. Resolva o PL Mestre;
4. Obtenha as variáveis duais π_i;
5. Gere uma coluna y para o PL Mestre resolvendo o subproblema (2.2) de forma exata;
6. **Se** $(1 - \sum_{i=1}^{m} \pi_i y_i < 0)$ **então**
7. Adicione a nova coluna o PL Mestre e volte ao passo 4;
8. Arredonde para baixo S^*_{Rel} para obter a solução inteira aproximada, S_{Apr} e $valorS_{Apr}$;
9. Determine os dados da instância residual PC_R: p_j`, $d_j$` e $itens_{PR}$;
10. Atualize $S^*_{PCU} = S_{Apr}$ e $valorS^*_{PCU} = valorS_{Apr}$;
11. **Se** $(itens_{PR} > 0)$ **faça**
12. Resolva o PC_R executando GR_{1d} (A, $iter_C$, $iter_R$, L, p_i',..., p_m', d_i',..., d_m');
13. Atualize $S^*_{PCU} = S^*_{PCU} \cup y_R$ e $valorS^*_{PCU} = valorS^*_{PCU} + Zy_R$;
14. Retorna (S^*_{PCU}, $valorS^*_{PCU}$);

Fim GCH_{1d}

Vale destacar que [Vieira Neto et al. 2013] não somente propuseram o algoritmo GRASP-1D mas deram destaque aos demais algoritmos híbridos ALG HB1e ALG HB2, que utilizam a fase de construção do GRASP-1D mas se diferenciam nas estratégias adotadas na fase posterior. Em comum, tem-se que os três algoritmos usam um gerador de padrões de corte nas instâncias PCU_{BL}, utilizando todos os padrões viáveis na estrutura de formulação matemática (2.1), para esta ser resolvida pelo método Simplex, relaxando a integridade de suas variáveis.

5. Resultados Computacionais

Os algoritmos GR_{1d} e GCH_{1d}, ambos para tratar o PCU, foram implementados em linguagem C/C++ (Visual C++ 2019), com as relaxações do PL Mestre e os subproblemas sendo resolvidos pelo CPLEX *Optimization Studio* 20.1.0, e os testes realizados em ambiente computacional com processador Intel(R) Core(TM) i7-1165G7 2.8 GHz, com 16GB de memória RAM e sistema operacional *Windows* 10 de 64 bits. Como não se tem informações precisas sobre a máquina utilizada nos testes dos algoritmos GRASP-1D, ALG HB1 e ALG HB2, foram considerados apenas os dados referentes ao valor de função objetivo.

Os resultados apresentados nos testes computacionais com os algoritmos propostos estão divididos em quatro grupos de instâncias reais, retiradas de fábricas, e indicadas conforme os dados dos projetos por Bg1, Bg2, Bg3 e Bg4. Segundo [Vieira Neto et al. 2013], o total de 35 instâncias não segue uma sequência padronizada de numeração, pois adotou-se a codificação original. Ressalta-se, também, que algumas instâncias foram excluídas por apresentarem dados incompletos ou incoerentes. Como por exemplo, no grupo Bg1, que vai de Bg1001 até Bg1010, falta a instância Bg1007.

TABELA 5.1 – Resultados do Algoritmo Híbrido GCH_{1d} nas Instâncias Práticas.

Código	m	*GC*	*GRASP-1D*		*ALG HB1*		*ALG HB2*		*GR1d*					*GCH1d*					
		Limite Inferior	Z	Gap%	Z	Gap%	Z	Gap%	Z	Zm	Tm(s)	Alfa	Gap%	Z Aprox	Z	Zm	Tm(s)	Cols	Gap%
Bg 1001	8	185,67	188	1,08	187	0,54	187	0,54	192	192	1,98	0,1	3,23	183	**186***	**186***	1,74	22	0
Bg 1002	14	209,70	212	0,95	212	0,95	**210***	0	215	215	3,00	0,3	2,38	207	**210***	**210***	1,30	34	0
Bg 1003	15	206,97	210	1,45	209	0,97	209	0,97	213	213	2,28	0,1	2,90	201	**207***	207,8	2,66	61	0
Bg 1004	9	54,25	**55***	0	**55***	0	**55***	0	56	56	2,92	0,1	1,82	52	**55***	**55***	1,77	26	0
Bg 1005	18	181,05	184	1,10	184	1,10	**182***	0	186	186	2,50	0,3	2,20	176	**182***	**182***	1,68	47	0
Bg 1006	8	145,50	147	0,68	147	0,68	**146***	0	147	147	2,68	0,1	0,68	144	**146***	**146***	0,93	18	0
Bg 1008	12	77,36	80	2,56	79	1,28	79	1,28	81	81	2,34	0,1	3,85	74	**78***	**78***	1,91	43	0
Bg 1009	10	30,00	31	3,33	31	3,33	**30***	0	32	32	2,76	0,1	6,67	28	**30***	**30***	1,64	14	0
Bg 1010	21	291,65	296	1,37	295	1,03	295	1,03	300	300	2,64	0,925	2,74	284	**292***	**292***	3,00	86	0
Bg 2001	28	1.522,81	1575	3,41	1550	1,77	1528	0,33	1607	1607,4	2,24	0,9	5,52	1517	**1523***	**1523***	2,48	80	0
Bg 2002	46	1.458,69	1525	4,52	1509	3,43	1485	1,78	1561	1561,1	2,43	0,9	6,99	1450	**1459***	**1459***	4,48	203	0
Bg 2003	23	671,85	679	1,04	679	1,04	679	1,04	680	680	2,99	0,8	1,19	664	**672***	**672***	143,58	64	0
Bg 2004	23	349,67	362	3,43	359	2,57	355	1,43	365	365	2,19	0,8	4,29	341	**350***	**350***	3,56	83	0
Bg 2005	23	340,29	344	0,88	344	0,88	342	0,29	346	346	2,32	0,7	1,47	335	**341***	**341***	3,05	73	0
Bg 2006	23	695,25	710	2,01	710	2,01	699	0,43	715	715	2,08	0,9	2,73	694	**696***	**696***	1,60	70	0
Bg 2007	23	365,57	368	0,55	367	0,27	367	0,27	370	370,4	2,61	0,95	1,09	363	**366***	**366***	2,14	77	0
Bg 2009	46	1.905,64	1940	1,78	1939	1,73	1934	1,47	1953	1953,5	2,82	0,975	2,47	1890	**1906***	**1906***	41,72	193	0
Bg 2010	23	731,13	734	0,27	734	0,27	**732***	0	736	736	2,38	0,8	0,55	727	**732***	**732***	1,92	61	0
Bg 3001	49	2.023,46	2107	4,10	2103	3,90	2050	1,28	2112	2112,3	2,49	1	4,35	2006	**2024***	**2024***	8,27	188	0
Bg 3002	68	1.987,01	2075	4,38	2066	3,92	2020	1,61	2087	2087	2,65	1	4,98	1968	**1988***	**1988***	15,90	257	0
Bg 3004	37	529,92	547	3,21	547	3,21	544	2,64	550	550,9	2,50	0,95	3,77	516	531	531	28,24	125	0,19
Bg 3005	34	510,36	531	3,91	526	2,94	523	2,35	534	534	2,33	0,7	4,50	495	**511***	**511***	5,13	150	0
Bg 3007	34	551,53	560	1,45	560	1,45	555	0,54	563	563	2,47	0,8	1,99	544	**552***	**552***	2,96	112	0
Bg3009	68	2.843,32	2880	1,27	2880	1,27	2874	1,05	2880	2881,3	3,19	0,9	1,27	2817	**2844***	**2844***	8,27	252	0
Bg 3010	37	1.084,52	1111	2,40	1108	2,12	1106	1,94	1111	1111	2,73	1	2,40	1069	**1085***	**1085***	16,23	127	0
Bg 3011	14	92,69	94	1,08	94	1,08	94	1,08	95	95	2,18	0,1	2,15	85	**93***	**93***	2,78	37	0
Bg 4001	67	2.165,31	2268	4,71	2260	4,34	2194	1,29	2272	2272	2,43	1	4,89	2147	**2166***	**2166***	22,80	261	0
Bg 4002	79	2.124,62	2230	4,94	2216	4,28	2168	2,02	2235	2235,9	2,77	0,975	5,18	2102	2126	2126	96,97	387	0,05
Bg 4003	43	1.004,41	1014	0,90	1014	0,90	1012	0,70	1012	1013,2	3,61	0,3	0,70	986	**1005***	**1005***	4,05	130	0
Bg 4004	47	594,13	614	3,19	612	2,86	606	1,85	617	617	2,36	1	3,70	580	**595***	**595***	10,35	207	0
Bg 4005	44	575,54	597	3,65	594	3,13	589	2,26	600	600,9	3,69	0,9	4,17	560	**576***	576,2	9,02	210	0
Bg 4006	44	941,24	977	3,72	965	2,44	953	1,17	979	979,9	3,12	0,975	3,93	926	**942***	**942***	12,56	216	0
Bg 4007	44	559,32	565	0,89	565	0,89	563	0,54	565	565	2,60	0,8	0,89	545	**560***	**560***	3,98	156	0
Bg 4010	47	1.212,92	1244	2,56	1241	2,31	1240	2,23	1244	1244,9	3,47	0,95	2,56	1197	1214	1214	35,79	209	0,08
Bg 4011	24	126,02	128	0,79	**127***	0	**127***	0	128	128	2,25	0,4	0,79	116	**127***	**127***	7,13	48	0
Média				2,22		1,85		1,01			2,63		3,00				14,62		0,01

Sabendo que os valores de função objetivo (Z) informam o total de objetos utilizados, nas colunas 10 e 16 da tabela 5.1, são apresentados os resultados da melhor solução obtida com

os respectivos algoritmos GR_{1d} e GCH_{1d}. Já nas colunas 11 e 17, têm-se o valor médio (Z_m) em 10 execuções. É possível verificar na última linha que a abordagem híbrida obteve o melhor gap médio, certificando 32 soluções ótimas. Além disso, as soluções apresentadas nas instâncias Bg3004, Bg4002 e Bg4010 possui 1 unidade a mais do limite inferior dado pela Geração de Colunas. Entretanto, os tempos bem inferiores do GR_{1d} lhe garante certa competitividade, conforme conferido na coluna 12, com T_m(s) se referindo ao tempo total médio de execução. Na coluna 13 destaca-se, também, o valor do parâmetro α responsável por gerar tal solução.

Para conseguir os referidos resultados, em todos os testes foram adotados os seguintes parâmetros com o GR_{1d}: sorteios estabelecidos a partir de semente aleatória; os conjuntos da fase de diversificação são estabelecidos como A = {0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1}; o conjunto com os valores constituído a partir do α_R na fase reativa como A' = {α_R, $\alpha_R \pm 0{,}025$, $\alpha_R \pm 0{,}05$}; como número máximo de iterações nas respectivas fases, $iter_C = 20$ para cada parâmetros α obtido em A e $iter_R = 20$ para cada um dos valores pertencentes a A', totalizando 300 soluções construídas; na fase de melhoria também adotou-se o total de 20 iterações para cada valor α de 0,1 a 1, com incremento de 0,05. É importante ressaltar que tal parametrização foi escolhida levando-se em consideração que os tempos de execução médios não deveriam ultrapassar 5 segundos nas instâncias utilizadas.

Nas colunas 4, 6 e 8 da tabela 5.1 são encontrados os valores das respectivas soluções dos algoritmos híbridos GRASP-1D, ALG HB1 e ALG HB2. As instâncias foram testadas com o GRASP-1D rodando 10 iterações e os demais 100 iterações cara cada valor do parâmetro α, variando de 0,5 a 1,0 com incremento de 0,05. As 7 soluções ótimas certificadas pelo ALG HB2 demonstraram sua eficiência nesse grupo de instâncias, com o ALG HB1 obtendo 2 e o GRASP-1D apenas 1 solução ótima. Numa comparação entre os mais promissores algoritmos, o GCH_{1d} supera o ALG HB1 em 33 das 35 instâncias e o ALG HB2 em 28 instâncias, sem perder alguma.

6. Conclusões e Perspectivas Futuras

Neste trabalho são apresentados os algoritmos GR_{1d} e GCH_{1d}, onde o primeiro consiste numa metaheurística GRASP Reativa inspirada nos trabalhos [Vieira Neto et al. 2010] e [Vieira Neto et al. 2013], e o segundo combina a técnica de Geração de Colunas e o GR_{1d} para resolver os PCU_R, análogo a abordagem de [Velasco e Uchoa 2015]. Os resultados foram satisfatórios e o GCH_{1d} obteve soluções melhores do que as obtidas com os algoritmos híbridos propostos [Vieira Neto et al. 2013], totalizando 28 das 35 instâncias práticas retiradas da indústria. Além disso, os tempos computacionais são bem razoáveis, com a média de tempo de execução total em 2,63 e 14,62 segundos nos 4 quatro grupos de instâncias.

É interessante notar que, em 30 das instâncias de teste, foi possível obter com o GCH_{1d}, nas 10 execuções, o valor médio igual ao limite inferior obtido na fase de geração de colunas, provando sua otimalidade. O desempenho deste algoritmo híbrido também se comprovou com o gap médio de aproximadamente 0,01%.

Como trabalhos futuros, pretende-se implementar outras estratégias de melhoria para a segunda fase da GRASP, utilizar outras técnicas, exatas e heurísticas, em outras ampliações dos algoritmos evidenciados, e aplicar tais procedimentos a outras variantes dos problemas de corte e empacotamento. Destaca-se, também, a criação de uma interface Web que viabilize a utilização desses algoritmos de otimização por qualquer usuário.

Sobre o bolsista de Iniciação Científica, João Vitor Bernardo Moreira da Silva, sua preparação para seguir a carreira acadêmica tem acontecido desde o Ensino Médio do IFF, quando cursava o Técnico em Eletrônica e já estava vinculado a projeto de pesquisa como bolsista do Programa Jovens Talentos/FAPERJ. Atualmente, está matriculado no 5º período do bacharelado em Engenharia de Controle e Automação, também no IFF, e apresentou boa maturidade, iniciativa e competência no trabalho que desenvolveu, enquanto bolsista voluntário e depois recebendo a bolsa de iniciação científica (PIBIC/IFF), no projeto "Geração de Colunas e Heurísticas para o Problema de Corte Unidimensional", implementando os algoritmos,

confirmando o potencial para investigação científica com as modificações sugeridas nas reuniões e, dessa forma, contribuindo significativamente para a pesquisa.

Agradecimentos

Os autores gostariam de agradecer ao Instituto Federal Fluminense por todo o suporte e pela bolsa de Iniciação Científica cedida ao estudante.

Referências

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. e 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.

Dantzig, G.B. e Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research*, v. 8, n. 1, p. 101-111.

Feo, T. A. e Resende, M. G. C. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operation Research Letters*, n. 8, p. 67-71.

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

Garey M. R. e Johson D. S. (1979). *Computers and Intractability: a guide to the theory of NP-Completeness*. New York: W. H. Freeman and Company.

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

Gilmore, P. e Gomory, R. (1963). A linear programming approach to the cutting-stock problem - Part II. *Operations Research*, v.11, p.863-888.

Marcotte, O. (1986). An instance of the cutting stock problem for which the rounding property does not hold. *Operations Research Letters*, v. 4, n. 5, p. 239-243.

Prais, M. e Ribeiro, C.C. (2000). Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment. *INFORMS Journal on Computing*, v. 12, n. 3, p. 164-176.

Rietz, J.; Scheithauer, G. e Terno, J. (2002). Tighter bounds for the gap and non-IRUP constructions in the one-dimensional cutting stock problem. *Optimization*, v. 51, n. 6, p. 927-963.

Rios, A. e Poldi, K. (2020). Problema de corte de estoque unidimensional com sobras aproveitáveis: solução via metaheurística GRASP. *TEMA (São Carlos)*, 21(3), 441-460.

Scheithauer, G. e Terno, J. (1995) A branch & bound algorithm for solving one-dimensional cutting stock problems exactly. *Applicationes Mathematicae*, v. 23, n. 2, p. 151-167.

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.

Vieira Neto, E., Velasco, A.S. e Paula Junior, G.G. (2013). Algoritmos Grasp Híbridos para o Problema de Corte Unidimensional. In: *Anais do XLV SBPO* - Simpósio Brasileiro de Pesquisa Operacional, Natal.

Velasco, A. S. e Uchoa, E. (2015). Geração de Colunas para o Problema de Corte de Estoque Bidimensional Guilhotinado. In: *Anais do XLVII SBPO* - Simpósio Brasileiro de Pesquisa Operacional, Ipojuca.

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.