LI Simpósio Brasileiro de Pesquisa Operacional
Limeira–SP, 02 a 06 de setembro de 2019

Formulações para o Problema Integrado de Dimensionamento de Lotes e Minimização de Padrões de Corte

Silva, E. M.
Departamento de Matemática Aplicada, UNESP-IBILCE
eduardo.msilva094@gmail.com

Melega, G. M.
Departamento de Engenharia de Produção, UFScar
gislainemelega@gmail.com

Araujo, S.A.
Departamento de Matemática Aplicada, UNESP-IBILCE
silvio.araujo@unesp.br

RESUMO

Este trabalho trata de forma integrada o Problema de Dimensionamento de Lotes e o Problema de Minimização de Padrões de Corte e tem como objetivo analisar duas formulações para esse problema. A primeira formulação é baseada em uma formulação estendida, proposta na literatura para o Problema de Corte de Estoque, e a outra é baseada em uma formulação compacta para o mesmo problema. Foram realizados testes considerando 80 instâncias geradas aleatoriamente. Ambos os modelos foram resolvidos pelo pacote CPLEX e analisados considerando diferentes fatores.

PALAVRAS CHAVE. Problema de Dimensionamento de Lotes, Problema de Minimização de Padrões de Corte, Modelagem Matemática.

Tópicos: Otimização Combinatória

ABSTRACT

This work aims to analyze two formulations for the integrated Lot-Sizing and Pattern Minimization problems. The first formulation is based on an extended formulation proposed in the literature for the Cutting Stock problem, and the second one is based on a compact formulation for the same problem. Tests are performed using 80 randomly generated instances. Both model are solved by an optimization package and they were analyzed by different factors.

KEYWORDS. Lot-Sizing Problem. Pattern Minimization Problem. Mathematical Modeling.

Paper topics: Combinatorial Optimization

https://proceedings.science/p/106746?lang=pt-br

1. Introdução

O Problema de Dimensionamento de Lotes (PDL) e o Problema do Corte de Estoque (PCE) vêm sendo estudados há mais de 60 anos. O PDL consiste em planejar a quantidade de itens a ser produzida em várias (ou única) máquinas, em cada período, ao longo de um horizonte de tempo finito, de modo a atender uma certa demanda e podendo estar sujeito a algumas limitações, como por exemplo, limitação de capacidade, tendo como objetivo otimizar uma função, que pode ser, minimizar custos de estocagem e produção, por exemplo. O PCE consiste basicamente cortar objetos disponíveis em estoque em itens menores a fim de satisfazer uma determinada demanda de itens e atingir um certo objetivo, que pode ser minimizar o custo de perda, minimizar o custo de objetos usados ou maximizar o lucro.

Boa parte dos avanços encontrados na literatura sobre esses dois problemas estão voltados em como resolvê-los de maneira separada, uma vez que cada problema é da classe NP-*Hard*. Com o avanço em teorias, melhorias em *hardwares* e *softwares*, e melhor entendimento do comportamento individual desses problemas no ambiente industrial, a combinação desses dois problemas vem recebendo maior atenção recentemente por parte de pesquisadores e indústrias. Na literatura, o problema é tratado como o Problema Integrado de Dimensionamento de Lotes e Corte de Estoque (PIDLCE) [Melega et al., 2018]. Vários autores vem mostrando bons resultados ao integrar ambos problemas, [Vanzela et al., 2017], [Gramani e França, 2006; Gramani et al., 2009].

Durante a resolução do PCE o custo do processo de corte pode ser afetado de diferentes maneiras, como por exemplo, pelo número de padrões de cortes usados para satisfazer a demanda dos itens. Em alguns problemas práticos, a posição das facas na máquina (ou lasers) precisa ser ajustada a cada vez que há uma troca de padrão de corte utilizado para cortar os objetos. Tais ajustes, em geral, interrompem a produção fazendo com que haja um custo, cada vez que troca-se o padrão de corte (*setup*). Em situações assim, para que haja redução no custo total, um plano de produção com poucos padrões de cortes é mais adequado [de Araujo et al., 2014]. Neste contexto, surge o problema de Minimização de Padrões (PMP), que caracteriza-se como o problema de cortar os objetos até que a demanda das peças seja satisfeita utilizando a menor quantidade possível de padrões de corte diferentes [McDiarmid, 1999].

Neste trabalho, propõem-se tratar de forma integrada o PDL e o PMP, denominado aqui como Problema Integrado de Dimensionamento de Lotes e Minimização de Padrões (PIDLMP). O interesse nesses tipos de problemas surge de suas aplicações práticas em ambientes industriais, como por exemplo, indústrias de papel, móveis e fibras de vidro.

Especificamente, este trabalho tem como objetivo analisar dois modelos para o PIDLMP. Um modelo é baseado na formulação de [Gilmore e Gomory, 1961, 1963] e o outro baseado na formualação compacta não-linear apresentada em [Vanderbeck, 2000], que também é explorado em [Yanasse et al., 2018]. O modelo baseado no de Gilmore e Gomory foi resolvido por um método heurístico baseado na geração de colunas e no pacote de otimização CPLEX, enquanto o modelo baseado no trabalho de Vanderbeck foi resolvido pelo CPLEX. Foram utilizadas 80 instâncias aleatórias para testes e os modelos foram examinados considerando diversos fatores.

Este trabalho está organizado da seguinte forma. Na Seção 2 são apresentados os modelos matemáticos, na Seção 3 encontram-se os métodos de solução dos modelos descritos na Seção 2, na Seção 3 encontram-se a descrição das instâncias usadas e os resultados computacionais obtidos. Por fim, na Seção 5 as conclusões e propostas futuras são apresentadas.

2. Definição do Problema e Modelos Matemáticos

https://proceedings.science/p/106746?lang=pt-br

No PIDLMP, dado um horizonte de planejamento finito de T períodos, $t = 1, ..., T$, e objetos em estoque de comprimento L. Em cada período um conjunto de itens de tamanho l_i, $i = 1, ..., m$ tem que ser cortado para atender uma certa demanda d_{it}. O PIDLMP se resume em decidir quanto, quando e como produzir os itens de tamanhos solicitados cortando os objetos disponíveis em estoque de forma que uma certa função objetivo seja otimizada. Neste trabalho, o nosso objetivo é minimizar os custos de estoque e *setups* gerados pelo padrão de corte.

Observa-se que, neste trabalho, a minimização da perda que surge durante o processo de corte não é considerada na função objetivo. O foco está em problemas nos quais os custos de *setup* dos padrões de corte e os custos de estocagem dos itens são mais críticos comparados ao custo das perdas geradas. Por exemplo, em situações de alta demanda, determinadas empresas podem estar focadas apenas com a produtividade da máquina desconsiderando as perdas que, em alguns casos, podem ser reutilizadas ou recicladas.

2.1. Modelo de Gilmore e Gomory Adaptado (GGA)

Uma extensão do modelo apresentado em [Gilmore e Gomory, 1961, 1963] para o problema de corte de estoque é dada a seguir.

Definição: Um padrão de corte, apresentado pela primeira vez [Gilmore e Gomory, 1961], é descrito pelo vetor $A_j = \{a_{1j}, ..., a_{ij}, ..., a_{mj}\}$, em que a_{ij} representa a quantidade de itens de comprimento l_i cortados de acordo com o padrão j. O padrão é válido se satisfaz a equação (1):

$$l_1 a_{1j} + l_2 a_{2j} + \ldots + l_m a_{mj} \leq L. \tag{1}$$

Com base nisso, considere os seguintes índices e parâmetros:

Índices:

- $t = 1, \ldots, T$: número de períodos no horizonte de planejamento;
- $j = 1, \ldots, N$: número de padrões de corte para os objetos em estoque;
- $i = 1, \ldots, m$: número dos tipos de itens a serem cortados.

Dados:

- L: comprimento do objeto;
- l_i: comprimento do item i;
- d_{it}: demanda do item i no período t, $t = 1, ..., T$;
- M_t: número suficientemente grande, a cada período t.

Parâmetros:

- s_{jt}: custo de *setup* do padrão de corte j no período t;
- h_{it}: custo para estocar o item i no período t;
- a_{ij}: quantidade de itens i cortados no padrão de corte j;
- K: limitante para a quantidade de padrões de cortes.

Variáveis de Decisão:

- x_{jt}: número de objetos em estoque cortados de acordo com o padrão j no período t;
- y_{jt}: 1 caso haja *setup* para o padrão de corte j no período t, e 0 caso contrário;
- I_{it}: quantidade de itens i estocados ao final do período t.

O PIDLMP pode, então, ser formulador por:

Modelo *GGA*

$$\textbf{Minimize} \sum_{t=1}^{T}\sum_{i=1}^{m} h_{it}I_{it} + \sum_{t=1}^{T}\sum_{j=1}^{K} s_{jt}y_{jt} \tag{2}$$

s.a:

$$x_{jt} \leq M_t y_{jt} \qquad \forall j,\ \forall t \tag{3}$$

$$I_{i,t-1} + \sum_{j=1}^{K} a_{ij}x_{jt} = d_{it} + I_{it} \qquad \forall i,\ \forall t \tag{4}$$

$$x_{jt} \in \mathbb{N},\ y_{jt} \in \{0,1\} \qquad \forall j,\ \forall t \tag{5}$$

$$I_{it} \geq 0 \qquad \forall i,\ \forall t \tag{6}$$

A função objetivo (2) minimiza o custo para estocar os itens e o custo de *setup* dos padrões de corte. A restrição (3) força a variável de *setup* y_{jt} ser 1 sempre que algum objeto é cortado de acordo com o padrão j, i.e. $x_{jt} > 0$, consequentemente um valor referente ao custo de preparo do padrão de corte é contabilizado na função objetivo. A equação (4) representa a restrição de balanceamento de estoque. As restrições (5) e (6) são o domínio das variáveis.

2.2. Modelo de Vanderbeck Adaptado (VBA)

O modelo a seguir é baseado no modelo apresentado em Vanderbeck [2000], que foi utilizado para o PMP. Considere:

Variáveis de Decisão:

- a_{ij}: quantidade de itens do tipo i cortados de acordo com o padrão de corte j.

Note que, em comparação ao modelo GGA, a_{ij} se torna uma variável de decisão e não mais um parâmetro do problema. Assim, o PIDLMP pode ser reformulado por:

Modelo *VBA*

$$\textbf{Minimize} \sum_{t=1}^{T}\sum_{i=1}^{m} h_{it}I_{it} + \sum_{t=1}^{T}\sum_{j=1}^{K} s_{jt}y_{jt} \tag{7}$$

S.a:

$$\sum_{i=1}^{m} l_i a_{ij} \leq L y_{jt} \qquad \forall j,\ \forall t \qquad (8)$$

$$x_{jt} \leq M_t y_{jt} \qquad \forall j,\ \forall t \qquad (9)$$

$$I_{i,t-1} + \sum_{j=1}^{K} x_{jt} a_{ij} = d_{it} + I_{it} \qquad \forall i,\ \forall t \qquad (10)$$

$$a_{ij} \in \mathbb{Z}_+,\ \ y_{jt} \in \{0,1\} \qquad \forall i,\ \forall j,\ \forall t \qquad (11)$$

$$I_{it} \geq 0 \qquad \forall i,\ \forall t \qquad (12)$$

$$x_{jt} \in \mathbb{Z}_+ \qquad \forall j,\ \forall t \qquad (13)$$

A função objetivo (7) minimiza o custo de estoque e o custo de *setups* nos padrões de corte. A restrição (8) limita a quantidade de itens a ser cortada de um objeto não deixando que a soma de seus tamanhos ultrapasse o comprimento do objeto. A restrição (9) força a variável de *setup* y_{jt} ser 1 quando algum objeto é cortado de acordo com o padrão j, i.e. $z_{jt} > 0$. A restrição (10) assegura que a demanda de cada item seja satisfeita. As restrições (11), (12) e (13) especificam o domínio das variáveis.

3. Métodos de Solução

A seguir, os métodos utilizados para resolver os modelos da Seção 2 serão apresentadas.

3.1. Método de solução para o modelo GGA

Este modelo foi resolvido pelo pacote CPLEX. Os parâmetros a_{ij} foram determinados através da estratégia de Geração de Colunas proposta em [Gilmore e Gomory, 1961], o que caracteriza este método de solução como heurístico.

Durante os testes inicias a resolução da relaxação linear do modelo GGA apresentou *tailling off* em todas as classes de instâncias utilizadas, nesses casos, não há garantias de que a solução da relaxação linear do modelo (2)-(6) seja um limitante inferior para o problema inteiro. Note ainda, que a variável x_{jt} não faz parte da função objetivo e portanto, durante a geração de colunas, o subproblema de *pricing* depende apenas da solução dual das restrições de demanda. Uma alternativa considerada para contornar esse problema foi de tomar a restrição (3) como uma igualdade. Observe que isso sempre vale em uma solução ótima da relaxação linear do problema (devido ao custo na função objetivo). Desse jeito, a variável binária y_{jt} pode ser rescrita como x_{jt}/M_t e então, a relaxação linear do modelo GGA é dada por:

$$\textbf{Minimize} \sum_{t=1}^{T}\sum_{i=1}^{m} h_{it} I_{it} + \sum_{t=1}^{T}\sum_{j=1}^{K} \frac{s_{jt} x_{jt}}{M_t} \qquad (14)$$

s.a:

$$I_{i,t-1} + \sum_{j=1}^{K} a_{ij} x_{jt} = d_{it} + I_{it} \qquad \forall i,\ \forall t \qquad (15)$$

$$x_{jt} \in \mathbb{R}^+,\ y_{jt} \in [0,1] \qquad \forall j,\ \forall t \qquad (16)$$

$$I_{it} \geq 0 \qquad \forall i,\ \forall t \qquad (17)$$

Note que no modelo (14)-(17) a variável x_{jt} faz parte da função objetivo, assim, o valor s_{jt}/M_t está sendo computado no problema da mochila durante a geração de colunas. Ao resolver as relaxações lineares considerando o problema (14)-(17) o *tailling off* em todas as instâncias testadas foi solucionado. Portanto, para o modelo GGA o valor de Z_{RL} foi considerado na solução do modelo (14)-(17) e Z_{IP} foi considerando como na solução do modelo (2)-(6).

3.2. Método de solução para o modelo VBA

Note que o Modelo VBA é uma extensão do modelo proposto em [Kantorovich, 1960], caracterizado pela inclusão de uma variável e um conjunto de restrições que caracterizam padrões de corte usados. Além disso, a adição dessa variável torna a restrição (10) não-linear. A estratégia a seguir usada em [Yanasse e Morabito, 2008], foi utilizada para linearizar a restrição (10). A ideia do método é "binarizar"a variável z_{jt}, e então "linearizar"o produto de uma variável binária com uma inteira. Então, defina

$$x_{jt} = \sum_{o=0}^{k_{jt}} 2^o n_{ojt} \qquad \forall j, \qquad \forall t \tag{18}$$

em que $n_{ojt} \in \{0,1\}$ e k_{ot} é o maior inteiro menor ou igual a $1 + \log_2 K_{jt}$, no qual K_{jt} é um limitante superior para x_{jt} (k_{jt} é o número de bits necessários para representar K_{jt}).

Agora, substituindo x_{jt} como definido na Equação (18) tem-se o produto $n_{ojt}a_{ijt}$ na restrição (10). Note ainda que, $f_i = \lfloor L/l_i \rfloor$ é um limitante superior para a_{ij}. Seja $z_{oijt} \in \mathbb{Z}_+$ a variável que será o produto de n_{ojt} e a_{ij}. Temos, então:

$$z_{oijt} \leq a_{ij} \qquad \forall o,\ \forall i,\ \forall j,\ \forall t \tag{19}$$
$$z_{oijt} \leq f_i n_{ojt} \qquad \forall o,\ \forall i,\ \forall j,\ \forall t \tag{20}$$
$$z_{oijt} \geq a_{ij} - f_i(1 - n_{ojt}) \qquad \forall o,\ \forall i,\ \forall j,\ \forall t \tag{21}$$

Dessa forma, as restrições do modelo VBA podem ser reescritas da seguinte maneira:

$$\sum_{i=1}^{m} l_i a_{ij} \leq L y_{jt} \qquad \forall o,\ \forall t \tag{22}$$
$$x_{jt} \leq M_t y_{jt} \qquad \forall j,\ \forall t \tag{23}$$
$$x_{jt} = \sum_{o=0}^{k_{jt}} 2^o n_{ojt}\ \forall j,\ \forall t \tag{24}$$
$$z_{oijt} \leq a_{ij} \qquad \forall o,\ \forall i,\ \forall j,\ \forall t \tag{25}$$
$$z_{oijt} \leq f_i n_{ojt} \qquad \forall o,\ \forall i,\ \forall j,\ \forall t \tag{26}$$
$$z_{oijt} \geq a_{ij} - f_i(1 - n_{ojt}) \qquad \forall o,\ \forall i,\ \forall j,\ \forall t \tag{27}$$
$$I_{i,t-1} + \sum_{j=1}^{K} \sum_{o=0}^{k_j} 2^o z_{oijt} = d_{it} + I_{it} \qquad \forall o,\ \forall i,\ \forall j,\ \forall t \tag{28}$$
$$a_{ij} \in \mathbb{Z}_+,\ \ y_{jt} \in \{0,1\} \qquad \forall i,\ \forall j,\ \forall t \tag{29}$$
$$I_{it} \geq 0 \qquad \forall i,\ \forall t \tag{30}$$
$$x_{jt} \in \mathbb{Z}_+ \qquad \forall j,\ \forall t \tag{31}$$
$$z_{oijt} \in \mathbb{Z}_+,\ n_{ijt} \in \{0,1\} \qquad \forall o,\ \forall i,\ \forall j,\ \forall t \tag{32}$$

As restrições (22) e (23) são as mesmas do modelo não-linear. A restrição (24) é responsável por "binarizar"a variável x_{jt}. As restrições (25)- (27) asseguram que z_{oijt} seja o produto $a_{ij}n_{ojt}$. A restrição (28) é o balanceamento de estoque após a linearização do modelo original, e as restrições (29)- (32) representam o domínio das variáveis do modelo.

Este modelo foi resolvido pelo pacote CPLEX.

4. Resultados Computacionais

Nesta seção são apresentadas as instâncias e os resultados computacionais obtidos com as formulações GGA e VBA. A seguir encontram-se descritos os procedimentos de escolhas de alguns parâmetros dos modelos e detalhes das implementações.

O parâmetro M_t foi estipulado inicialmente da seguinte maneira: em cada instância, seja l_{Max} o comprimento da maior peça, assim, define-se:

$$M_1 = \left\lceil \frac{\sum_{i=1}^{m}\sum_{t=1}^{T} d_{it}}{l_{Max}} \right\rceil, \tag{33}$$

em que $\lceil . \rceil$ é a função teto. A fim de obter melhores limitantes através da relaxação linear dos problemas, esse parâmetro foi mudado a cada período da seguinte forma: no primeiro período usa-se o valor da Equação (33), e então para $t \in (1, T]$, $M_t = M_{t-1} - \left\lceil \frac{\sum_{i=1}^{m} d_{i(t-1)}}{l_{Max}} \right\rceil$.

Para cada instância utilizada o parâmetro K foi obtido da seguinte maneira: devido ao método de solução heurístico o valor de K para o modelo GGA é dado pela quantidade de padrões de corte gerados pela geração de colunas do problema, enquanto que, no modelo VBA, o parâmetro foi considerado como o dobro da quantidade de padrões de cortes utilizados na solução inteira do modelo GGA.

Os testes foram rodados no Visual Studio 2012 e resolvidos através do CPLEX versão 12.61. A máquina utilizada possui as seguintes configurações Intel Core i7 da quarta geração com 16GB de memória ram e 1000 GB de HD. O limite de tempo para cada instância foi de 3600 segundos para ambos os modelos.

4.1. Instâncias

[Poldi e de Araujo, 2016] propuseram um gerador de instâncias. Porém, esse processo gera mais de um tamanho de objeto por instância devido ao tipo de problema abordado pelo autores e além disso, não utiliza custos de *setup*, portanto foram adotados os valores como no trabalho de [Cui et al., 2015], que tomou como constantes para todas as instâncias com valores 1000 e 100 para L e s_{jt}, respectivamente. O restante dos dados foram gerados de acordo com os seguintes critérios: que foi adaptado para a classe de problemas tratada neste trabalho.

- Número de períodos: T=3, T=6;
- quantidade de itens: $m = 10$ ou $m = 20$;
- custo de estoque: $h_{it} = 0,01 * l_i$;
- tamanho dos itens: $l_i \in [0,1; 0,4]L$;
- demandas: $d_{it} \in [1, 50]$.

Foram utilizadas 4 classes de instâncias propostas por [Poldi e de Araujo, 2016], cada classe contém 20 instâncias. A Tabela 1 apresenta uma descrição geral de cada classe gerada.

Tabela 1: Descrição das classes

Classe	Quantidade de Períodos	Quantidade de Itens
1	3	10
2	3	20
3	6	10
4	6	20

4.2. Resultados

Uma avaliação por médias dos fatores avaliados nos modelos em cada classe se encontra nas Tabelas 2 e 3. Porém, os resultados do Modelo VBA para as classes 2, 3 e 4 não foram bons e por isso serão apresentados apenas os dados obtidos pela sua relaxação linear. A fim de fornecer uma melhor percepção do comportamento dos modelos uma avaliação individual em cada instância da Classe 1 foi feita e está apresentada na Tabela 4. Todos os dados estão com aproximações de até 3 casas decimais.

Com relação aos parâmetros usados na comparação dos modelos, a avaliação foi feita com base no valor da relaxação linear do problema (Z_{RL}), tempo para resolver a relaxação linear em segundos, (T_{RL}), valor da solução inteira (Z_{PI}), tempo usado para resolver o problema inteiro em segundos, (T_{PI}), quantidade de objetos usados na solução inteira (O) e a quantidade de padrões utilizados na solução inteira (*setups*), S.

Tabela 2: Resultados do Modelo GGA.

Modelo Adaptado Gilmore e Gomory						
Classe	Z_{RL}	T_{RL}	Z_{IP}	T_{IP}	O	S
2	$110,382$	$0,143$	$2743,675$	$2023,377$	265	19
3	$147,0123$	$0,098$	$2726,857$	$2890,951$	268	18
4	$132,431$	$0,270$	$6396,587$	3600	556	41

Tabela 3: Resultados do Modelo VBA.

Modelo Adaptado Vanderbeck		
Classe	Z_{RL}	T_{RL}
2	$69,415$	$4,124$
3	$110,8492$	$7,586$
4	$87,534$	$105,708$

Analisando os dados das Tabelas 2, 3 e 4 pode-se destacar as seguintes conclusões:

- **Relaxação linear e tempo de resolução:** nota-se que tanto nas médias das classes $2, 3$ e 4 quanto nos resultados por instâncias da classe 1 o modelo GGA apresentou melhores limitantes para o problema inteiro comparado ao modelo VBA. Ainda, a resolução da relaxação linear para o modelo de Vanderbeck exigiu um maior esforço computacional em todos os casos. O maior esforço computacional está relacionado com a técnica de linearização descrita na Subseção 3.2 .

- **Problema inteiro e tempo de resolução:** pela Tabela 4 nota-se que o modelo GGA apresentou maior média de valor inteiro, mais especificamente isso se deve ao fato de o modelo ter

Tabela 4: Resultados dos modelos GGA e VBA para classe de instâncias 1.

	Modelo GGA						**Modelo VBA**					
Instância	Z_{RL}	T_{RL}	Z_{PI}	T_{PI}	O	S	Z_{RL}	T_{RL}	Z_{PI}	T_{PI}	O	S
1	112,511	0,062	1304,96	31,265	120	10	82,48	2	1316,77	3600	198	8
2	100,131	0,094	1071,1	1,29688	142	8	58,743	2,453	1367,23	3600	192	10
3	95,916	0,094	1355,76	24,0313	141	10	56,465	2,5	1265,89	3600	197	9
4	128,062	0,094	1337,84	8,4375	138	9	94,015	1,703	1237,78	3600	190	8
5	115,153	0,062	1342,68	2,89063	131	9	84,349	2,047	1176,77	3600	151	7
6	101,82	0,062	1563,84	2,75	148	11	59,892	2,64	1350,88	3600	184	8
7	122,376	0,141	1361,31	48,657	131	9	89,658	1,828	1062,7	3600	151	6
8	106,721	0,109	1360,55	26,531	155	8	62,653	2,36	1479,61	3600	185	7
9	124,159	0,109	1366,64	13,516	135	9	90,995	1,937	1070,76	3600	159	9
10	109,367	0,141	1475,94	4,578	155	10	64,395	2,328	1498,43	3600	191	13
11	125,858	0,062	1248,77	1,672	135	9	92,276	1,828	1326,97	3600	148	10
12	106,912	0,094	1610,71	1,016	155	11	62,865	2,61	1388,64	3600	259	10
13	102,564	0,094	1341,39	8,812	114	10	75,056	1,719	1225,87	3600	178	8
14	114,721	0,125	1384,59	18,047	120	10	84,156	2,016	1385,92	3600	217	9
15	94,700	0,078	1397,77	10,828	143	9	55,793	2,203	1245,67	3600	149	10
16	128,17	0,094	1322,26	3,860	136	9	93,9	1,75	1387,29	3600	190	11
17	121,041	0,078	1237,81	9,094	129	9	88,641	2,11	1271,68	3600	167	9
18	127,483	0,109	1377,08	25,172	139	9	93,556	2,203	1107,57	3600	213	8
19	113,103	0,094	1339,19	7,062	124	9	82,946	2,031	916,43	3600	133	5
20	121,415	0,1094	1326,5	1,91	130	9	89,176	1,969	1211,01	3600	157	7
Média	113,61	0,096	1356,334	12,572	136	10	78,16	2,112	1265,802	3600	180	9

obtido um valor maior em 14 instâncias da classe 1. Como era de se esperar, no geral, visto que o modelo GGA foi resolvido por um método heurístico. Ainda, não tem-se garantia da solução ótima de nenhuma instância resolvida pelo VBA, visto que todas rodaram pelo tempo máximo estipulado.

- **Quantidade de objetos e *setups* :** aqui percebe-se uma relação entre os valores da função objetivo e o número de padrões de corte utilizados na solução. O modelo GGA foi o que mais utilizou padrões de cortes, em geral, nas soluções (em 13 instâncias) e consequentemente, teve uma maior solução na função objetivo do problema inteiro. Em contrapartida a quantidade de objetos cortados pelo modelo VBA foi maior que a do modelo GGA em todas as instâncias da classe 1.

5. Conclusões e Propostas Futuras

Neste trabalho, foram examinadas formulações para o Problema Integrado de Dimensionamento de Lotes e Minimização de Padrões. Foram utilizados dois modelos, uma extensão baseada no modelo apresentado em [Gilmore e Gomory, 1961, 1963] e uma extensão baseada no trabalho de [Vanderbeck, 2000]. Instâncias aleatórias foram geradas para o problema e então realizou-se os testes computacionais.

Em termos de valor objetivo, para a classe 1 (instâncias pequenas), o modelo VBA apresentou melhores resultados comparados aos obtidos pelo modelo GGA no tempo máximo estipulado para os testes. No entanto, se mostrou um modelo "pesado"em termos de esforço computacional para as classes maiores, isso se deve ao fato da linearização inserir muitas variáveis no problema.

Uma proposta de trabalho futuro é adicionar à função objetivo um custo de perda afim de minimizar a quantidade de objetos utilizados, além de resolver o problema de simetria apontado por [Vanderbeck, 2000], isto é, o problema dado pela variável j dos padrões de corte não seguirem uma ordenação natural na solução.

Outro fator relevante não considerando na análise dos modelos foi o *gap* da solução inteira. Isso se deve ao fato desse valor estar muito elevado em todos os testes devido a baixa qualidade dos limitantes inferiores fornecidos pela relaxação linear dos problemas. Outra proposta futura é estudar maneiras de melhorar o valor da relaxação a fim de obter melhores limitantes, visto que no problema de corte de estoque em uma dimensão o valor da relaxação linear do problema é um limitante de alta qualidade para o problema inteiro.

Referências

Cui, Y., Zhong, C., e Yao, Y. (2015). Pattern-set generation algorithm for the one-dimensional cutting stock problem with setup cost. *European Journal of Operational Research*, 243(2):540 – 546.

de Araujo, S. A., Poldi, K. C., e Smith, J. (2014). A genetic algorithm for the one-dimensional cutting stock problem with setups. *Pesquisa Operacional*, 34:165 – 187.

Gilmore, P. C. e Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6):849 – 859.

Gilmore, P. C. e Gomory, R. E. (1963). A linear programming approach to the cutting stock problema - part ii. *Operations Research*, 11(6):863 – 888.

Gramani, M. C. N. e França, P. M. (2006). The combined cutting stock and lot-sizing problem in industrial processes. *European Journal of Operational Research*, 174(1):509 – 521.

https://proceedings.science/p/106746?lang=pt-br

Gramani, M. C. N., França, P. M., e Arenales, M. N. (2009). A lagrangian relaxation approach to a coupled lot-sizing and cutting stock problem. *International Journal of Production Economics*, 119(2):219 – 227.

Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management Science*, 6(4):366 – 422.

McDiarmid, C. (1999). Pattern minimisation in cutting stock problems. *Discrete Applied Mathematics*, 98(1-2):121 – 130.

Melega, G. M., de Araujo, S. A., e Jans, R. (2018). Classification and literature review of integrated lot-sizing and cutting stock problems. *European Journal of Operational Research*, 271(1):1 – 19.

Poldi, K. C. e de Araujo, S. A. (2016). Mathematical models and a heuristic method for the multiperiod one-dimensional cutting stock problem. *Annals of Operations Research*, 238(1):497 – 520.

Vanderbeck, F. (2000). Exact algorithm for minimising the number of setups in the one-dimensional cutting stock problem. *Operations Research*, 48(6):915–926.

Vanzela, M., Melega, G. M., Rangel, S., e de Araujo, S. A. (2017). The integrated lot sizing and cutting stock problem with saw cycle constraints applied to furniture production. *Computers & Operations Research*, 79:148 – 160.

Yanasse, H. H. e Morabito, R. (2008). A note on linear models for two-group and three-group two-dimensional guillotine cutting problems. *International Journal of Production Research*, 46(21): 6189–6206.

Yanasse, H. H., Neto, L. L. S., e Costa, C. R. (2018). Uma abordagem exata para o problema de corte biobjetivo. In *Caderno de resumos da XIX ONPCE- Oficina Nacional de Corte, Empacotamento, Planejamento e Programaçao de Produção e Correlatos*, Unesp, Bauru-SP.