Modelo para o Problema de Carregamento em um único Contêiner com Restrições de Agrupamento e Fragilidade

Oliviana Xavier do Nascimento e Thiago Alves de Queiroz
Unidade de Matemática e Tecnologia - UFG/Regional Catalão.
Av. Dr. Lamartine Pinto de Avelar, 1120, Setor Universitário, 75704-020, Catalão-GO, Brasil.
olivianaxn@gmail.com
taq@ufg.br

RESUMO

O tema do artigo é o desenvolvimento de um método exato para resolver o Problema de Carregamento em um único Contêiner. O método é constituído por uma relaxação, um modelo de programação por restrições e inserção de cortes. Os cortes são inseridos toda vez que o subconjunto de itens escolhido pela relaxação não pode ser empacotado dentro do contêiner. O modelo de programação por restrições decide se os itens escolhidos podem ou não ser empacotados. Além disso, considera-se a inserção das restrições de agrupamento de itens e itens frágeis. O método é codificado em linguagem C++ e usa as bibliotecas do pacote de otimização da *IBM ILOG CPLEX Optimization Studio*. Experimentos computacionais realizados em instâncias da literatura para validar o método mostram que as piores soluções em termo de ocupação são obtidas quando a restrição de agrupamento de itens é considerada.

PALAVRAS CHAVE. Problema de Carregamento em Contêineres. Agrupamento de itens. Restrição de fragilidade.

Tópicos: OC - Otimização Combinatória. IND – PO na Indústria

ABSTRACT

The theme of this paper is the development of an exact method to solve the single Container Loading Problem. The method consists of a relaxation, a constraint programming model and the insertion of cuts. The cuts are inserted whenever a subset of items chosen by the relaxation cannot be packed inside the container. And, the constraint programming model decides whether the chosen items can be packed. In addition, it is considered the insertion of the constraints complete shipment of items and the load bearing strength of items. The method is coded in the C++ programming language and it uses some libraries of the IBM ILOG CPLEX Optimization Studio. Computational experiments performed on instances of the literature to validate the method show that the worst solutions in terms of occupied volume are obtained when the complete shipment constraint is considered.

KEYWORDS. Container Loading Problem. Complete shipment. Load bearing.

Paper Topics: OC - Combinatorial Optimization. IND - OR in Industry.

1. Introdução

Este trabalho desenvolve um método exato para resolver o Problema de Carregamento em um único Contêiner (PCumC) com e sem as restrições de agrupamento de itens e fragilidade. O PCumC busca selecionar, dentre um conjunto de itens (caixas), um subconjunto ótimo de itens para ser empacotado em um único recipiente (contêiner) [Wäscher et al., 2007]. O subconjunto de itens é ótimo quando possui valor máximo dentre todos os subconjuntos disponíveis e que podem ser empacotados. O subconjunto pode ser empacotado quando seus itens ocupam, cada um, posições diferentes dentro do recipiente, não há sobreposição entre itens, todos os itens estão inteiramente contidos no recipiente e os itens são organizados de forma a ter seus lados paralelos aos lados do recipiente (empacotamento ortogonal).

O PCumC surge no contexto de muitas indústrias. Ele é considerado um problema importante no setor logístico das mesmas, pois a sua resolução garante que o recipiente contenha a maior ocupação possível, quando o valor é tido como volume do item, diminuindo, por conseguinte, os gastos desnecessários com alocação de novos recipientes ou veículos de carga.

A principal restrição envolvendo o PCumC é a restrição que visa garantir a não sobreposição dos itens. No entanto, há outros tipos de restrições que podem ser associadas a este problema. Tais restrições visam tornar o problema mais próximo do que ocorre na realidade de empresas e indústrias. Bischoff e Ratcliff [1995] apresentaram alguns exemplos de restrições que podem ser associadas a problemas de empacotamento, dentre elas, as restrições de agrupamento de itens e a de fragilidade.

A restrição de agrupamento de itens leva em consideração a existência de clientes para os quais os itens devem ser entregues. A ideia é selecionar quais serão os clientes que terão os seus itens empacotados no recipiente. Isto pressupõe que, se um item de um determinado cliente for empacotado, os demais itens deste cliente devem ser empacotados também [Eley, 2003]. Esta restrição surge na prática quando os clientes solicitam peças para o conserto de máquinas e há apenas um veículo para servir a demanda, o que torna necessário escolher entre quais clientes atender com os itens.

A restrição de fragilidade está associada com o empacotamento de itens marcados como frágeis. Cada item frágil possui uma limitação quanto ao número de outros itens que podem ser empacotados acima dele e exercendo alguma influência, mesmo que indiretamente [Bruns et al., 2016]. Esta restrição impõe a condição de que a quantidade de itens sobre cada item frágil não deve exceder a quantidade permitida pelo item frágil. As aplicações envolvendo fragilidade são inúmeras e surgem no carregamento de caixas com limite no número de outras caixas que podem vir acima. Assim, a consideração de fragilidade evita possíveis danos físicos e estruturais nas mercadorias transportadas.

O método adotado neste trabalho para resolver o PCumC relaxa o problema, resolvendo um modelo de Programação Inteira (PI) do tipo mochila unidimensional e depois resolve um problema de viabilidade relacionado ao empacotamento tridimensional. A relaxação do problema faz com que seja selecionado um subconjunto tal que o volume total dos itens não ultrapasse o volume do recipiente e que, ao mesmo tempo, maximize o valor do empacotamento. Apesar de ser um processo efetivamente mais rápido, não há garantia de que os itens do subconjunto selecionado não se sobreponham quando colocados dentro do recipiente. Logo, torna-se necessário checar se existe alguma maneira de empacotar os itens sem que eles se sobreponham e ainda fiquem totalmente contidos no recipiente. Isto envolve resolver o problema de empacotamento ortogonal tridimensional, denominado por 3OPP.

A resolução do 3OPP envolve um modelo que é resolvido usando a técnica de Programação por Restrições (PR). Se a resolução deste modelo resultar que não há como empacotar o subconjunto selecionado, o modelo relaxado é chamado fazendo a inserção de um corte para que, assim, seja selecionado um subconjunto diferente. Esse novo subconjunto precisa ser checado pelo modelo de PR e o processo continua até que haja algum subconjunto ótimo de itens para o PCumC.

O método proposto também considera a inserção das restrições de agrupamento de itens e fragilidade. Quando a restrição de agrupamento de itens é considerada, faz-se a sua inserção na relaxação do problema. Desse modo, além de respeitar as requisições de volume, o subconjunto selecionado deve satisfazer as condições impostas pela restrição de agrupamento de itens. Quando a restrição de fragilidade é considerada, faz-se a sua inserção no modelo para o 3OPP. Com isso, busca-se decidir se existe uma configuração de empacotamento para o subconjunto que, além da condição de não sobreposição, empacotamento ortogonal e de os itens estarem contidos no recipiente, satisfaça as condições de fragilidade.

Ao relaxar o PCumC, o número de variáveis cai quando comparado ao número requerido para tratar o problema diretamente na versão tridimensional [Queiroz et al., 2017]. Isso faz com que a resolução da relaxação seja rápida utilizando PI. Ao resolver o 3OPP na PR, ganha-se com as técnicas de redução de domínio, que contribuem para agilizar a resolução do problema de checar a viabilidade do empacotamento. Com base nisso, o método deste trabalho busca unir as vantagens de se utilizar a PI e a PR na resolução do PCumC. Os experimentos deste trabalho buscam validar o método apresentado para resolver o PCumC e investigar o impacto da inserção das restrições de agrupamento de itens e fragilidade neste problema. Com isso, investiga-se se há aumento no tempo de resolução ou perca de volume empacotado ao considerar tais restrições.

A organização deste trabalho se dá da seguinte maneira. Na seção 2, apresenta-se a revisão de alguns trabalhos que trataram o PCumC, bem como das restrições de agrupamento de itens e fragilidade. Esta seção também menciona os trabalhos que serviram de referência para o método desenvolvido aqui ou que trazem métodos parecidos com ele. A seção 3 traz o método para resolver o PCumC, apresentando, portanto, o modelo relaxado e o de programação por restrições que o método utiliza. Na seção 4 são definidas as restrições de agrupamento de itens e fragilidade. Nessa seção ainda é mencionado em que ponto do método elas são inseridas. A seção 5 explica os experimentos computacionais realizados a fim de validar o método desenvolvido e comentar sobre o impacto das restrições consideradas na solução final do PCumC. E, por fim, na seção 6 são feitas as conclusões do trabalho e indicadas direções para trabalhos futuros.

2. Revisão da Literatura

Wäscher et al. [2007], em seu trabalho sobre a tipologia dos problemas de corte e empacotamento, classificaram o PCumC de acordo com a heterogeneidade dos itens. Para casos em que os itens são muito diferentes entre si (muito heterogêneos), o PCumC é classificado como um problema da mochila. Já para casos em que os itens são pouco diferentes entre si (pouco heterogêneos), o PCumC é classificado como um *Single Large Placement Packing Problem* (3-SLOPP). Ambas classificações são consideradas NP-difíceis [Garey e Johnson, 1979]. Por este motivo, a literatura tem recorrido mais aos métodos heurísticos, porque, embora sejam mais rápidos do que os métodos exatos, costumam fornecer boas soluções para o problema [Bortfeldt e Wäscher, 2013].

Algumas heurísticas para o PCumC buscam empacotar os itens por meio de camadas verticais. A exemplo, tem-se as heurísticas que foram propostas em George e Robinson [1980] e Bortfeldt e Gehring [2001]. Assim, George e Robinson [1980] buscaram manter as camadas bem próximas umas das outras para evitar espaços vazios e uma consequente perda de volume empacotado. Outras heurísticas, como a apresentada em Bischoff e Ratcliff [1995], buscam empacotar os itens por meio de camadas horizontais. Há ainda heurísticas que seguem o critério de empacotar os itens por meio de blocos contendo itens do mesmo tipo [Eley, 2002].

Em relação a métodos exatos para o PCumC, alguns exemplos podem ser encontrados em Martello et al. [2000], Hifi [2004] e Junqueira et al. [2012]. Martello et al. [2000] propuseram um algoritmo *branch-and-bound*. Hifi [2004] trabalhou com um algoritmo de programação dinâmica, ao passo que Junqueira et al. [2012] propuseram um modelo de programação linear inteira resolvido por um algoritmo *branch-and-cut*.

Abordagens semelhantes a que é proposta neste trabalho para resolver o PCumC foram apresentadas por Caprara e Monaci [2004], Baldacci e Boschetti [2007] e Queiroz et al. [2017] para o Problema da Mochila Bidimensional (2KP). Estes autores propuseram relaxar o 2KP para selecionar um subconjunto de itens e depois utilizaram uma estratégia para empacotar os itens dentro do recipiente. O que diferencia o método para o PCumC tratado aqui das abordagens destes autores é que eles utilizaram programação linear inteira para checar como empacotar os itens dentro do recipiente, a menos de [Queiroz et al., 2017], que, como aqui, empregaram a técnica de programação por restrições.

Alguns trabalhos também utilizaram a programação por restrições para resolver o problema de checar como arranjar todos os itens de um conjunto dentro de um único recipiente. Como exemplo, tem-se Pisinger e Sigurd [2007], Clautiaux et al. [2008], Mesyagutov et al. [2012a], Mesyagutov et al. [2012b] e Nascimento et al. [2016]. Estes trabalhos foram usados como referenciais para a definição do modelo de programação por restrições utilizado aqui. Nesses trabalhos, os autores definiram um modelo básico de PR e, a partir dele, construíram o próprio algoritmo para resolver o problema ou usaram alguma biblioteca já publicada. Para construir os algoritmos, os autores buscaram por métodos para agregar novas restrições ao modelo básico. A intenção com novas restrições é tirar maior proveito dos conceitos envolvidos na PR, neste caso, a propagação de restrições.

Alguns trabalhos que trataram da restrição de agrupamento de itens são os de Eley [2003] e Queiroz et al. [2017]. Eley [2003] propôs uma heurística e Queiroz et al. [2017] propuseram uma abordagem heurística e outra exata. Em ambos trabalhos, os autores trataram também da existência de itens conflitantes, ou seja, itens que não podem ser empacotados no mesmo recipiente. Em Queiroz et al. [2017], se o algoritmo com conflitos decide remover um item de um cliente, os demais itens deste cliente também são removidos. Caso o algoritmo decida incluir um item de um cliente, os demais itens deste cliente devem ser carregados, o que é exigido pela restrição de agrupamento de itens.

A restrição de fragilidade pode ser encontrada, dentre outros trabalhos, nos de Scheithauer e Terno [1997], Bischoff [2006], Egeblad et al. [2010], Junqueira et al. [2012], Ceschia e Schaerf [2013], Queiroz e Miyazawa [2013], Bruns et al. [2016], Le e Knust [2017]. Bischoff [2006], Junqueira et al. [2012], Ceschia e Schaerf [2013] e Queiroz e Miyazawa [2013] consideraram que cada caixa permite uma pressão máxima em sua face superior e, então, assumiram que a pressão total acima de toda caixa frágil não deve ultrapassar a pressão permitida. Bruns et al. [2016] e Le e Knust [2017] consideraram diretamente a quantidade de itens que cada caixa frágil suporta, ao invés da pressão. Scheithauer e Terno [1997] optaram por proibir que caixas maiores fossem colocadas acima de caixas menores. De maneira semelhante, Egeblad et al. [2010] e Le e Knust [2017] optaram por proibir que caixas mais pesadas fossem colocadas acima de caixas mais leves.

3. Método para o PCumC

Uma instância do PCumC considera um conjunto $S = \{1, 2, \ldots, m\}$ de itens, em que cada item i de S possui largura l_i, comprimento c_i, altura a_i e valor v_i, e um contêiner de largura L, comprimento C e altura A. O objetivo é determinar um subconjunto de S cujo valor total do empacotamento seja máximo e que os itens caibam todos de forma ortogonal e sem sobreposição dentro do contêiner.

Para não existir sobreposição entre os itens do subconjunto selecionado, deve existir um ponto $p_i = (X_i, Y_i, Z_i)$ dentro do recipiente onde cada item i do subconjunto é empacotado sem que o item intercepte pontos cobertos por outros itens. O sistema de referências adotado para os itens e para o recipiente está em termo dos eixos x, y e z, em que o eixo x está associado à largura, o eixo y ao comprimento e o eixo z à altura.

3.1. Relaxação para o problema

O método proposto para resolver o PCumC considera, inicialmente, a resolução do modelo apresentado na formulação (1)-(3), que é um modelo de programação linear inteira conhecido para resolver o problema da mochila 0-1. Este modelo consiste em uma relaxação para o PCumC, pois se limita a resolver a versão unidimensional do problema, a qual não considera questões geométricas para lidar com a sobreposição de itens. A variável de decisão utilizada nessa formulação é z_i, que assume valor 1 caso o item i seja colocado no recipiente e valor 0, caso ocorra o contrário, para todo $i = 1, 2, \ldots, m$.

$$\max \sum_{i=1}^{m} v_i z_i \tag{1}$$

Sujeito a:

$$\sum_{i=1}^{m} (l_i c_i a_i) z_i \leq LCA, \tag{2}$$

$$z_i \in \{0, 1\}, \quad i = 1, \ldots, m. \tag{3}$$

Na formulação (1)-(3), a função objetivo (1) representa a solução de máximo valor total, dado os itens empacotados no recipiente. A restrição (2) garante que o volume total referente aos itens empacotados não exceda o volume do recipiente. Por fim, o domínio das variáveis é dado nas restrições (3), neste caso indicando variáveis binárias. Com esta formulação, um subconjunto U de máximo valor e que respeita o volume do recipiente é selecionado. Note que não é possível afirmar que a solução encontrada é viável para o PCumC sem antes checar a sua viabilidade quanto ao empacotamento.

3.2. Modelo de programação por restrições

Para resolver o problema de checar a viabilidade de empacotar os itens do subconjunto U que foi selecionado pelo modelo da mochila 0-1, define-se o modelo (4). Ele é composto por uma única restrição, que é a de não sobreposição, e ela é definida para todo par de itens i e j. O modelo possui como variáveis de decisão X_i, Y_i e Z_i, que representam o ponto $p_i = (X_i, Y_i, Z_i)$ onde cada item i do subconjunto selecionado é empacotado no recipiente.

$$X_i + l_i \leq X_j \vee X_j + l_j \leq X_i \vee Y_i + c_i \leq Y_j \vee Y_j + c_j \leq Y_i \vee Z_i + a_i \leq Z_j \vee \quad Z_j + a_j \leq Z_i \tag{4}$$

O modelo (4) verifica se para todo par de itens i e j de U existe, respectivamente, um ponto p_i e p_j onde estes itens podem ser empacotados sem que eles se sobreponham. Tal modelo pode ser escrito e resolvido usando uma formulação por programação inteira, porém neste trabalho ele é resolvido usando programação por restrições.

Assume-se no modelo (4) que o domínio das variáveis de decisão X_i, Y_i e Z_i é obtido, sem perda de generalidade, sobre os conjuntos X_d, Y_d e Z_d dados em (5), (6) e (7), respectivamente. Esses conjuntos são obtidos pela estratégia de discretização *canonical dissections* de Herz [1972]. Segue que o domínio das variáveis de decisão do item i consiste nesses conjuntos ao retirar as coordenadas inválidas para posicioná-lo dentro do recipiente.

$$X_d = \{p \in \mathbb{Z} \mid p = \sum_{i=1}^{m} \varepsilon_i l_i, 0 \leq p \leq L - \min_{1 \leq i \leq m} \{l_i\}, \quad 0 \leq \varepsilon_i \leq 1, \varepsilon_i \in \mathbb{Z}, i = 1, \ldots, m\} \tag{5}$$

$$Y_d = \{q \in \mathbb{Z} \mid q = \sum_{i=1}^{m} \varepsilon_i c_i, 0 \leq q \leq C - \min_{1 \leq i \leq m} \{c_i\}, \quad 0 \leq \varepsilon_i \leq 1, \varepsilon_i \in \mathbb{Z}, i = 1, \ldots, m\} \tag{6}$$

$$Z_d = \{r \in \mathbb{Z} \mid r = \sum_{i=1}^{m} \varepsilon_i a_i, 0 \leq r \leq A - \min_{1 \leq i \leq m} \{a_i\}, \quad 0 \leq \varepsilon_i \leq 1, \varepsilon_i \in \mathbb{Z}, i = 1, \ldots, m\} \tag{7}$$

3.3. Cortes

Se o modelo de programação por restrições, que considera a restrição (4), verificar que existe uma configuração de empacotamento viável para os itens de U, então se tem uma solução ótima para o PCumC. Caso ocorra o contrário, um corte, a restrição (8), é inserido na formulação do problema da mochila 0-1 para que, assim, outro subconjunto U_1 (diferente de U) seja obtido por este modelo com a restrição (8). Segue que o modelo de programação por restrições é chamado para verificar a viabilidade de U_1. Esses passos são repetidos até que o modelo de programação por restrições encontre um empacotamento viável dado o subconjunto retornado pela formulação do problema da mochila 0-1 com cortes do tipo (8).

$$\sum_{i \in U} z_i \leq |U| - 1 \tag{8}$$

A desigualdade em (8) evita que seja obtido um subconjunto de itens igual ao U, uma vez que ele não atende a viabilidade do empacotamento. Isso faz com que a formulação (1)-(3) selecione um subconjunto diferente de máximo valor (tal valor é sempre menor ou igual ao de U).

4. Restrições Práticas

Visando atender alguns critérios necessários ao carregamento de caixas em contêineres, que são comuns em empresas do ramo logístico, considera-se as restrições práticas de agrupamento e fragilidade de itens.

4.1. Agrupamento de itens

Considere que existem n clientes para os quais os itens devem ser entregues e que A_k representa o conjunto que contém os itens que devem ser entregues ao cliente k, para todo $k = 1, \ldots, n$. A restrição em (9), inserida na formulação (1)-(3), impõe que se um item do cliente k for escolhido para o empacotamento, todos os demais itens deste cliente devem ser empacotados também.

$$|A_k| z_i \leq \sum_{j \in A_k} z_j, \quad \forall i \in A_k, \quad k = 1, \ldots, n. \tag{9}$$

4.2. Fragilidade

A restrição de fragilidade busca garantir que itens frágeis não sejam danificados por excesso de outros itens colocados sobre eles. Para modelar esta restrição, considere F como o conjunto de todas as caixas frágeis de U e f_i como a quantidade de caixas que podem ser empacotadas sobre uma caixa i tida como frágil. A ideia é fazer com que não mais do que f_i caixas sejam colocadas de forma a ter a sua base projetada sobre a face superior da caixa i. Para isto, considere a variável de decisão auxiliar b_{ij}. Ela assume 1 se a caixa j é empacotada acima da caixa i e tem a sua base projetada sobre a face superior de i gerando alguma interseção, e valor 0, caso ocorra o contrário. A modelagem desta restrição considera que uma caixa j está acima de uma caixa i se a projeção de sua base, sobre os eixos x e y, intercepta a face superior de i. Assim, a restrição (10) garante que, sobre cada caixa i frágil, podem existir no máximo f_i caixas.

$$\sum_{j=1}^{|U|} b_{ij} \leq f_i, \quad \forall i \in F. \tag{10}$$

5. Experimentos Computacionais

O método para o PCumC foi codificado na linguagem de programação C++. Realizaram-se três tipos de experimentos computacionais sobre instâncias adaptadas da literatura para verificar a validade do método, a saber:

- Experimento 1: O PCumC foi resolvido na sua forma básica, sem as restrições práticas.

- Experimento 2: O PCumC foi resolvido considerando a restrição de agrupamento de itens.

- Experimento 3: O PCumC foi resolvido considerando a restrição de fragilidade.

Os experimentos ocorreram em um computador com processador Intel Core i5-3570 de 3,4 GHz, 8 GB de memória RAM e sistema Linux Ubuntu 14.04 LTS. Para a resolução dos modelos foi utilizada as bibliotecas do pacote *IBM ILOG CPLEX Optimization Studio 12.6.1* [IBM, 2014].

As instâncias para os experimentos foram adaptadas de Junqueira et al. [2012], considerando que a demanda de cada item é unitária. Assim, não há comparação direta com os resultados de Junqueira et al. [2012] ou de outros autores da literatura. Em particular, tem-se que Junqueira et al. [2012] não consideraram a restrição referente ao agrupamento de itens e, com relação a restrição de fragilidade, tais autores a propuseram de uma maneira diferente da que foi proposta aqui. Em Junqueira et al. [2012], a restrição de fragilidade considera que cada caixa permite uma pressão máxima em sua face superior e, neste trabalho, tal restrição foi proposta considerando que cada caixa permite uma certa quantidade de itens acima dela.

Em cada experimento há instâncias com 10 e 20 itens e recipientes de lados iguais, com volumes 10^3, 20^3 e 30^3, totalizando 60 instâncias. Para cada combinação entre quantidade de itens e volume de recipiente, tem-se um total de 10 instâncias. Os itens das instâncias são considerados grandes em relação ao tamanho do recipiente e foi estabelecido o tempo limite de 3600 segundos para resolução de cada instância. No Experimento 2, considerou-se 3 clientes nas instâncias com 10 itens e, nas com 20 itens, considerou-se 4 clientes. No experimento 3, considerou-se que 30% dos itens são frágeis.

5.1. Resultados

Na Tabela 1 encontram-se os resultados para as instâncias com 10 e 20 itens. As instâncias na Tabela 1 se diferenciam quanto ao volume do recipiente, que pode ser 10^3, 20^3 ou 30^3. Nesta tabela são apresentados o nome da instância, a descrição do experimento, que pode ser sem restrições práticas, com a restrição de agrupamento de itens ou com a restrição de fragilidade. No nome de cada instância, N está associado ao tamanho do recipiente e m está associado à quantidade de itens. Para cada experimento e instância é apresentado o número de cortes inseridos durante a resolução, o último valor assumido pela função objetivo, se o método conseguiu encontrar uma solução ótima no tempo limite e o tempo gasto, em segundos, para encontrar a solução. Caso o método não tenha conseguido encontrar uma solução ótima no tempo limite, o campo solução é marcado com “-”.

Para as instâncias com 10 itens, o método com e sem restrições práticas conseguiu resolver todas as instâncias sem ultrapassar o tempo limite. Houve diminuição do valor da função objetivo para 80% das instâncias deste grupo quando se considerou a restrição de agrupamento e, para 70% das instâncias, quando se considerou a restrição de fragilidade. Houve também diminuição no tempo médio de resolução e no número médio de cortes para todos os tamanhos de recipiente quando se considerou a restrição de agrupamento. No caso em que se considerou a restrição de fragilidade, houve aumento no tempo médio de resolução e no número de cortes para todos os tamanhos recipiente, exceto para o de tamanho 30.

Para as instâncias com 20 itens, o método sem restrições práticas e com a restrição de fragilidade encontrou dificuldades durante a resolução, pois não conseguiu encontrar qualquer solução ótima em 3600 segundos para as instâncias. Para estes casos, o que há é um limite superior para o valor da solução ótima. Já no experimento em que se considerou a restrição de agrupamento foi possível encontrar a solução ótima para aproximadamente 77% das instâncias.

Tabela 1: Resultado para as instâncias dado os diferentes experimentos.

Instância	Experimento 1 (Sem restrições práticas)				Experimento 2 (Com agrupamento)				Experimento 3 (Com fragilidade)			
	Cortes	V.F.O	Solução	Tempo (s)	Cortes	V.F.O	Solução	Tempo (s)	Cortes	V.F.O	Solução	Tempo (s)
N10m10b2d2-01	18	698	Ótima	0,91	1	566	Ótima	0,10	122	578	Ótima	3,13
N10m10b2d2-02	0	486	Ótima	0,01	0	486	Ótima	0,01	0	486	Ótima	0,04
N10m10b2d2-03	0	610	Ótima	0,02	0	610	Ótima	0,02	0	610	Ótima	0,04
N10m10b2d2-04	0	592	Ótima	0,02	0	592	Ótima	0,01	0	592	Ótima	0,03
N10m10b2d2-05	0	613	Ótima	0,01	0	613	Ótima	0,01	85	488	Ótima	0,73
N10m10b2d2-06	0	650	Ótima	0,01	0	650	Ótima	0,02	0	650	Ótima	0,13
N10m10b2d2-07	0	555	Ótima	0,01	0	555	Ótima	0,02	0	555	Ótima	0,07
N10m10b2d2-08	8	667	Ótima	0,89	2	452	Ótima	0,18	36	607	Ótima	2,66
N10m10b2d2-09	26	611	Ótima	1,42	2	503	Ótima	0,14	26	611	Ótima	1,07
N10m10b2d2-10	13	682	Ótima	0,19	1	568	Ótima	0,02	13	682	Ótima	0,44
N20m10b2d2-01	553	4792	Ótima	21,21	4	3390	Ótima	0,08	620	4418	Ótima	25,12
N20m10b2d2-02	13	6241	Ótima	0,11	2	4756	Ótima	0,02	107	5233	Ótima	1,25
N20m10b2d2-03	231	5910	Ótima	3,51	4	3226	Ótima	0,01	453	4902	Ótima	9,79
N20m10b2d2-04	125	6060	Ótima	22,16	1	5262	Ótima	0,02	306	5083	Ótima	7,01
N20m10b2d2-05	29	5632	Ótima	2,05	1	4664	Ótima	0,08	173	4574	Ótima	12,26
N20m10b2d2-06	67	6304	Ótima	1,99	0	6054	Ótima	0,01	212	5352	Ótima	5,81
N20m10b2d2-07	63	5894	Ótima	5,57	2	5168	Ótima	0,18	236	4912	Ótima	4,19
N20m10b2d2-08	440	5441	Ótima	6,55	2	2781	Ótima	0,07	578	4629	Ótima	11,65
N20m10b2d2-09	22	6663	Ótima	1,31	2	5397	Ótima	0,11	27	6509	Ótima	1,45
N20m10b2d2-10	363	5478	Ótima	11,69	3	2781	Ótima	0,01	468	5004	Ótima	17,99
N30m10b2d2-01	311	19062	Ótima	9,16	3	11795	Ótima	0,01	311	19062	Ótima	9,77
N30m10b2d2-02	134	19418	Ótima	99,07	2	15711	Ótima	0,25	258	17402	Ótima	12,23
N30m10b2d2-03	404	16132	Ótima	10,96	2	15834	Ótima	0,02	470	15424	Ótima	12,50
N30m10b2d2-04	460	17998	Ótima	10,01	4	7612	Ótima	0,06	468	17838	Ótima	10,86
N30m10b2d2-05	408	18306	Ótima	12,28	2	13626	Ótima	0,01	430	17722	Ótima	14,60
N30m10b2d2-06	323	17141	Ótima	28,59	2	15283	Ótima	0,12	456	15419	Ótima	13,41
N30m10b2d2-07	30	20358	Ótima	2,41	4	11820	Ótima	0,27	72	18684	Ótima	9,67
N30m10b2d2-08	350	17930	Ótima	16,65	4	11073	Ótima	0,07	374	17519	Ótima	0,03
N30m10b2d2-09	357	19314	Ótima	6,07	4	6738	Ótima	0,02	357	19314	Ótima	6,21
N30m10b2d2-10	85	19840	Ótima	7,10	1	17338	Ótima	0,11	186	17709	Ótima	15,57
N10m20b2d2-01	3170	994	–	3600,00	50	808	Ótima	5,31	3023	994	–	3600,00
N10m20b2d2-02	133	999	–	3600,00	1	995	–	3600,00	3517	974	–	3600,00
N10m20b2d2-03	96	996	–	3600,00	47	834	Ótima	287,67	4030	996	–	3600,00
N10m20b2d2-04	1	1000	–	3600,00	0	994	–	3600,00	1351	974	–	3600,00
N10m20b2d2-05	4127	980	–	3600,00	31	888	Ótima	4,73	1800	990	–	3600,00
N10m20b2d2-06	128	998	–	3600,00	12	894	–	3600,00	3597	960	–	3600,00
N10m20b2d2-07	8	998	–	3600,00	1	970	–	3600,00	908	980	–	3600,00
N10m20b2d2-08	2132	980	–	3600,00	20	892	Ótima	28,27	401	996	–	3600,00
N10m20b2d2-09	5685	822	–	3600,00	122	512	Ótima	1,01	4771	828	–	3600,00
N10m20b2d2-10	18	1000	–	3600,00	0	1000	–	3600,00	4268	962	–	3600,00
N20m20b2d2-01	1707	7989	–	3600,00	39	5740	Ótima	0,81	1683	7989	–	3600,00
N20m20b2d2-02	1712	7985	–	3600,00	39	5525	Ótima	0,54	1689	7985	–	3600,00
N20m20b2d2-03	1245	7993	–	3600,00	28	6656	Ótima	5,43	1278	7993	–	3600,00
N20m20b2d2-04	1482	7991	–	3600,00	50	5509	Ótima	0,98	1447	7991	–	3600,00
N20m20b2d2-05	1639	7985	–	3600,00	56	4583	Ótima	0,76	1619	7985	–	3600,00
N20m20b2d2-06	1506	7992	–	3600,00	16	6665	Ótima	0,35	1462	7992	–	3600,00
N20m20b2d2-07	1367	7991	–	3600,00	83	3002	Ótima	2,32	1482	7990	–	3600,00
N20m20b2d2-08	1340	7994	–	3600,00	46	6016	Ótima	1,43	1408	7993	–	3600,00
N20m20b2d2-09	296	7999	–	3600,00	26	7047	–	3600,00	1385	7992	–	3600,00
N20m20b2d2-10	1412	7994	–	3600,00	30	6790	–	3600,00	1408	7994	–	3600,00
N30m20b2d2-01	1379	26973	–	3600,00	42	21565	Ótima	0,70	1352	26974	–	3600,00
N30m20b2d2-02	1373	26966	–	3600,00	53	16448	Ótima	2,04	1369	26966	–	3600,00
N30m20b2d2-03	1754	26923	–	3600,00	41	15952	Ótima	1,21	1724	26925	–	3600,00
N30m20b2d2-04	1502	26965	–	3600,00	75	11829	Ótima	0,77	1477	26965	–	3600,00
N30m20b2d2-05	1519	26960	–	3600,00	66	13911	Ótima	0,68	1496	26961	–	3600,00
N30m20b2d2-06	1376	26968	–	3600,00	75	12630	Ótima	1,66	1368	26968	–	3600,00
N30m20b2d2-07	1342	26963	–	3600,00	59	15158	Ótima	2,50	1312	26964	–	3600,00
N30m20b2d2-08	1489	26964	–	3600,00	50	19085	Ótima	1,35	353	26992	–	3600,00
N30m20b2d2-09	1536	26958	–	3600,00	46	16732	Ótima	47,92	1514	26958	–	3600,00
N30m20b2d2-10	1533	26962	–	3600,00	61	16918	Ótima	1,36	1495	26964	–	3600,00

Houve diminuição no valor da função objetivo para, aproximadamente, 97% das instâncias quando a restrição de agrupamento foi considerada e, para 27% das instâncias, quando se considerou a restrição de fragilidade. Houve também diminuição no tempo médio de resolução e no número médio de cortes, para todos os tamanhos de recipiente, quando se considerou a restrição de agrupamento. No caso em que se considerou a restrição de fragilidade, observou-se uma diminuição no número médio de cortes para o grupo de instâncias com recipiente de tamanho 30.

Ainda com relação a esta restrição, observou-se uma diminuição no tempo médio de resolução para o grupo de instâncias com recipiente de tamanho 10.

A Tabela 2 sintetiza os resultados da Tabela 1. Nela são apresentados, para cada combinação entre quantidade de itens e recipiente, o valor mínimo, médio e máximo obtido com relação à quantidade de cortes inseridos, valor da função objetivo e tempo de resolução. Cada combinação entre uma quantidade m de itens e um tamanho N de recipiente representa um grupo de 10 instâncias da Tabela 1.

Tabela 2: Síntese dos resultados apresentados na Tabela 1.

Nº de Itens	Recipiente	Tipo	Experimento 1			Experimento 2			Experimento 3		
			Cortes	F.O.	Tempo (s)	Cortes	F.O.	Tempo (s)	Cortes	F.O.	Tempo (s)
10	10^3	Mínimo	0	486	0,01	0	452	0,01	0	486	0,03
		Média	6,50	616,40	0,35	0,60	559,50	0,05	28,20	585,90	0,83
		Máximo	26	698	1,42	2	650	0,10	122	682	3,13
	20^3	Mínimo	13	4792	0,11	0	2781	0,01	27	4418	1,25
		Média	190	5841,50	7,62	2,10	4448	0,06	31,80	5061,60	9,65
		Máximo	553	666,30	22,16	4	6054	0,18	620	6509	25,12
	30^3	Mínimo	30	16132	2,41	1	6738	0,01	72	15419	0,03
		Média	286,20	18549,90	20,23	2,80	12683	0,09	338,20	17609,30	10,48
		Máximo	460	20358	99,07	4	17338	0,27	470	19314	15,57
20	10^3	Mínimo	1	822	3600	0	512	1,01	401	828	3600,00
		Média	5249,80	976,70	3600,00	28,40	878,70	1832,70	2766,60	965,40	3600,00
		Máximo	41127	1000	3600,00	122	1000	3600,00	4771	996	3600,00
	20^3	Mínimo	296	7985	3600,00	16	3002	0,35	1278	7985	3600,00
		Média	1370,60	7991,30	3600,00	41,30	5753,30	721,26	1486,10	7990,40	3600,00
		Máximo	1712	7999	3600,00	83	7047	3600,00	1689	7994	3600,00
	30^3	Mínimo	1342	26923	3600,00	41	11829	0,68	353	26925	3600,00
		Média	1480,30	26960,20	3600,00	56,80	16022,80	6,02	1346	26963,70	3600,00
		Máximo	1754	26973	3600,00	75	21565	47,92	1726	26992	3600,00

A diminuição mínima, média e máxima do valor da função objetivo com a inserção das restrições práticas é informada na Tabela 3, para todos os tamanhos de recipiente e conforme a quantidade de itens presente nas instâncias.

Tabela 3: Redução em porcentagem (%) no valor da função objetivo dada a inserção das restrições práticas.

Recipiente	Restrição de agrupamento (Experimento 2)						Restrição de fragilidade (Experimento 3)					
	Instâncias com 10 itens			Instâncias com 20 itens			Instâncias com 10 itens			Instâncias com 20 itens		
	Mínima	Média	Máxima	Mínima	Média	Máxima	Mínima	Média	Máxima	Mínima	Média	Máxima
10^3	0,00	32,23	8,55	0,40	11,70	37,71	0,00	20,39	4,66	1,80	2,68	3,81
20^3	3,97	23,67	48,89	11,90	28,01	62,43	2,31	13,88	18,79	0,01	0,04	0,09
30^3	1,85	31,11	65,11	20,05	40,57	56,13	0,00	5,01	10,74	-	-	-

A Tabela 4 apresenta, para cada tamanho de recipiente, a diminuição observada no número médio de cortes e no tempo médio de resolução quando se considera a restrição de agrupamento de itens. Já a Tabela 5 apresenta, para cada tamanho de recipiente, se houve aumento (+) ou diminuição (-) no número médio de cortes e no tempo de resolução quando se considera a restrição de fragilidade e quanto houve de aumento ou diminuição.

Tabela 4: Diminuição em porcentagem (%) dado o número médio de cortes e o tempo médio de resolução para a restrição de agrupamento de itens.

Recipiente	Instâncias com 10 itens		Instâncias com 20 itens	
	Cortes	Tempo	Cortes	Tempo
10^3	90,77	97,42	99,46	49,09
20^3	98,90	99,21	96,87	79,09
30^3	99,02	99,54	96,14	99,83

Perceba, na Tabela 5, que houveram casos em que se verificou aumento no número de cortes ao mesmo tempo em que se verificou diminuição no tempo médio de resolução. Com base nisso e levando em consideração que a restrição de fragilidade é inserida no modelo de PR, pode-se dizer que, nestes casos, o modelo de PR foi rápido em verificar a viabilidade dos subconjuntos selecionados pelo modelo da mochila.

Tabela 5: Diminuição (-) ou aumento (+) em porcentagem (%) dado o número médio de cortes e o tempo médio de resolução para a restrição de fragilidade.

Recipiente	Instâncias com 10 itens		Instâncias com 20 itens	
	Cortes	Tempo	Cortes	Tempo
10^3	333,85 (+)	237,82 (+)	99,46 (+)	47,30 (-)
20^3	83,26 (+)	126,74 (+)	8,43 (+)	0,00
30^3	18,17 (+)	40,67 (-)	9,07 (-)	0,00

6. Considerações Finais e Trabalhos Futuros

Neste trabalho foi desenvolvido um método exato para resolver o Problema de Carregamento em um Único Contêiner (PCumC). O método é composto de uma relaxação do problema da mochila 0-1 unidimensional, um modelo de programação por restrições para resolver o problema de empacotamento ortogonal tridimensional e a inserção de cortes na relaxação, que são oriundos de soluções inviáveis do modelo de programação por restrições. Investigou-se também o que acontece com o número de cortes, valor da função objetivo (que está associado com o volume empacotado) e o tempo de resolução quando as restrições práticas de agrupamento de itens e fragilidades são consideradas no PCumC.

Os experimentos computacionais indicaram que o método para o problema sem restrições práticas e com a restrição de fragilidade encontrou dificuldades em resolver instâncias com 20 itens, enquanto que instâncias com 10 itens foram resolvidas rapidamente, em alguns casos, em menos de 1 segundo. O aumento do tempo computacional à medida que a quantidade de itens aumenta, deve-se ao aumento do número de variáveis.

Os experimentos também mostraram que considerar o problema com a restrição de agrupamento de itens contribuiu para encontrar a solução ótima para mais instâncias dentro do tempo limite de 3600 segundos quando comparado aos demais casos investigados. Além disso, o número médio de cortes inseridos durante a resolução, bem como o tempo médio de resolução, foram menores do que quando se considera o problema sem nenhuma restrição prática. Nota-se, ainda, aumento no número médio de cortes e no tempo médio de resolução quando a restrição de fragilidade é considerada. Verificou-se, ainda, que as maiores percas, com relação a volume empacotado, foram obtidas quando a restrição de agrupamento de itens foi considerada.

Como trabalho futuro desta pesquisa, busca-se considerar a inclusão de outras restrições práticas, como a estabilidade de carga e conflito entre os itens em problemas de empacotamento. Outra linha está relacionada ao desenvolvimento e uso de novas restrições visando melhorar o desempenho do modelo de programação por restrições.

Agradecimentos

Os autores agradecem o apoio financeiro recebido da Coordenadoria de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - processo 308312/2016-3) e da Fundação de Amparo à Pesquisa do Estado de Goiás (FAPEG).

Referências

Baldacci, R. e Boschetti, M. A. (2007). A cutting-plane approach for the two-dimensional orthogonal non-guillotine cutting problem. *European Journal of Operational Research*, 183(3):1136–1149.

Bischoff, E. E. e Ratcliff, M. (1995). Issues in the development of approaches to container loading. *Omega*, 23(4):377–390.

Bischoff, E. (2006). Three-dimensional packing of items with limited load bearing strength. *European Journal of Operational Research*, 168(3):952–966.

Bortfeldt, A. e Gehring, H. (2001). A hybrid genetic algorithm for the container loading problem. *European Journal of Operational Research*, 131(1):143–161.

Bortfeldt, A. e Wäscher, G. (2013). Constraints in container loading–a state-of-the-art review. *European Journal of Operational Research*, 229(1):1–20.

Bruns, F., Knust, S., e Shakhlevich, N. V. (2016). Complexity results for storage loading problems with stacking constraints. *European Journal of Operational Research*, 249(3):1074–1081.

Caprara, A. e Monaci, M. (2004). On the two-dimensional knapsack problem. *Operations Research Letters*, 32(1):5–14.

Ceschia, S. e Schaerf, A. (2013). Local search for a multi-drop multi-container loading problem. *Journal of Heuristics*, 19(2):275–294.

Clautiaux, F., Jouglet, A., Carlier, J., e Moukrim, A. (2008). A new constraint programming approach for the orthogonal packing problem. *Computers & Operations Research*, 35(3):944 – 959.

Egeblad, J., Garavelli, C., Lisi, S., e Pisinger, D. (2010). Heuristics for container loading of furniture. *European Journal of Operational Research*, 200(3):881–892.

Eley, M. (2002). Solving container loading problems by block arrangement. *European Journal of Operational Research*, 141(2):393–409.

Eley, M. (2003). A bottleneck assignment approach to the multiple container loading problem. *OR Spectrum*, 25(1):45–60.

Garey, M. R. e Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, San Francisco.

George, J. A. e Robinson, D. F. (1980). A heuristic for packing boxes into a container. *Computers & Operations Research*, 7(3):147–156.

Herz, J. (1972). Recursive computational procedure for two-dimensional stock cutting. *IBM Journal of Research and Development*, 16(5):462–469.

Hifi, M. (2004). Exact algorithms for unconstrained three-dimensional cutting problems: a comparative study. *Computers & Operations Research*, 31(5):657–674.

IBM (2014). Ilog cplex optimization studio v12.6.3 documentation. `https://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.1/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.html`. Acesso em 06 de março de 2017.

Junqueira, L., Morabito, R., e Yamashita, D. S. (2012). Three-dimensional container loading models with cargo stability and load bearing constraints. *Computers & Operations Research*, 39(1): 74–85.

Le, X. T. e Knust, S. (2017). Mip-based approaches for robust storage loading problems with stacking constraints. *Computers & Operations Research*, 78:138–153.

Martello, S., Pisinger, D., e Vigo, D. (2000). The three-dimensional bin packing problem. *Operations Research*, 48(2):256–267.

Mesyagutov, M., Scheithauer, G., e Belov, G. (2012a). Lp bounds in various constraint programming approaches for orthogonal packing. *European Journal of Operational Research*, 39(10): 2425–2438.

Mesyagutov, M., Scheithauer, G., e Belov, G. (2012b). New constraint programming approaches for 3d orthogonal packing. *Technische Universität Dresden, Dresden, Germany*.

Nascimento, O. X., Cunha, J. G. A., e Queiroz, T. A. (2016). Resolução do problema de empacotamento ortogonal com diferentes malhas e restrições reais. *Pesquisa Operacional para o Desenvolvimento*, 8(3):236–265.

Pisinger, D. e Sigurd, M. (2007). Using decomposition techniques and constraint programming for solving the two-dimensional bin-packing problem. *INFORMS Journal on Computing*, 19(1): 36–51.

Queiroz, T. A. e Miyazawa, F. K. (2013). Two-dimensional strip packing problem with load balancing, load bearing and multi-drop constraints. *International Journal of Production Economics*, 145(2):511–530.

Queiroz, T. A., Hokama, P. H. D. B., Schouery, R. C. S., e Miyazawa, F. K. (2017). Two-dimensional disjunctively constrained knapsack problem: Heuristic and exact approaches. *Computers & Industrial Engineering*, 105:313–328.

Scheithauer, G. e Terno, J. (1997). A heuristic approach for solving the multi-pallet packing problem. *Decision making under conditions of uncertainty (cutting and packing problems)*, p. 140–154.

Wäscher, G., Haußner, H., e Schumann, H. (2007). An improved typology of cutting and packing problems. *European journal of operational research*, 183(3):1109–1130.