

# Exact Method for the Container Loading Problem with Priority and Stability

1<sup>st</sup> Oliviana Xavier do Nascimento

3<sup>rd</sup> Thiago Alves de Queiroz

*Instituto de Matemática e Tecnologia*

*Universidade Federal de Goiás - Regional Catalão*

Catalão-GO, Brasil

olivianaxn@gmail.com

taq@ufg.br

2<sup>nd</sup> Luciana Vieira de Melo

*Programa de Pós-Graduação em Engenharia de Produção*

*Universidade Federal de Goiás - Regional Catalão*

Catalão-GO, Brasil

luciana\_vieira92@hotmail.com

**Abstract**—In this work, it is presented an exact method to solve the Single Container Loading Problem, which aims at deciding which items should be packed into the container in order to attain the maximum loaded value. The method is derived from the resolution of integer and constraint programming models. One of the integer programming models is a relaxation for the three-dimensional packing problem related to arranging items into the container. Besides that, it is measured how constraints related to priority conditions and vertical and horizontal load cargo stability can impact on the solution. Instances from the literature are solved in order to assess the method behavior in relation to the occupied volume and runtime. It has been observed that the highest volume losses have happened when the priority conditions are taken into consideration, while the runtime has increased when the vertical and horizontal stability constraints are considered.

**Index Terms**—Container loading problem; mathematical programming models; load priority; cargo stability.

## I. INTRODUÇÃO

Este trabalho aborda o Problema de Carregamento em único Contêiner (PCumC). Tal problema é bastante comum em empresas que lidam com transporte e armazenamento de itens em recipientes, como contêineres e baús de caminhões. Os itens no PCumC são caixas, e tanto estas quanto o recipiente possuem formato retangular. O objetivo do problema é decidir quais caixas carregar no contêiner para que o valor do carregamento seja máximo [1]. Além disso, busca-se determinar as coordenadas para posicionar as caixas escolhidas de tal forma que as caixas não extrapolem as dimensões do recipiente e de forma que não ocorra sobreposição de caixas no leiaute de empacotamento gerado, isto é, nenhuma caixa pode ocupar posições cobertas por outras caixas no empacotamento resultante.

Dado que o PCumC é um problema prático, em [2] há a descrição de restrições, algumas ainda pouco abordadas pela literatura existente, mas que são importantes de serem consideradas quando se resolve problemas de carregamento de itens em contêineres. Tais restrições capturam aspectos relacionados aos contextos reais de aplicação do problema, tornando a solução encontrada mais coerente com as demandas desses contextos. Restrições para garantir a estabilidade do

carregamento e para garantir a prioridade de envio de certas caixas são alguns exemplos. Outros exemplos são restrições para tratar questões associadas a fragilidade dos itens, descarregamento e entrega, e carregamento completo dos itens de um mesmo cliente.

Diante disso, este trabalho apresenta uma estratégia para resolver o PCumC, que envolve modelos de programação inteira (PI) e programação por restrições (PR). Os modelos são resolvidos iterativamente usando o solver IBM ILOG CPLEX até que a solução ótima do problema seja encontrada. Um dos modelos consiste em uma relaxação para o Problema de Empacotamento Ortogonal Tridimensional (3OPP), que surge como subproblema no PCumC no momento de buscar por coordenadas para posicionar os itens dentro do contêiner. Neste caso, o objetivo pretendido ao resolver o 3OPP é checar se todos os itens podem ser posicionados de forma que eles não se sobreponham, fiquem dispostos de forma ortogonal às paredes do contêiner e inteiramente dentro do contêiner.

Abordagens semelhantes a adotada neste trabalho para resolver o PCumC podem ser encontradas em [3] e [4], porém, aplicadas ao problema da mochila bidimensional, que pode ser visto como a versão bidimensional do PCumC. Outro detalhe que convém destacar é que [3] e [4] não fizeram uso da relaxação baseada na mochila bidimensional. Sobre esta relaxação, não se encontrou trabalhos na literatura que fizeram uso ou menção. Assim, tal relaxação se configura como uma das contribuições que esse trabalho traz.

Além disso, o presente trabalho apresenta como modelar e incorporar nos modelos duas restrições adicionais, a saber, restrição de prioridade e estabilidade de carga vertical e horizontal. A restrição de prioridade faz-se necessária quando há no conjunto caixas com diferentes prioridades de envio. Neste caso, a restrição atua garantindo que as caixas de maior prioridade estejam presentes no carregamento. Já a restrição de estabilidade vertical, conhecida também por restrição de estabilidade estática, garante que as caixas não se desloquem com relação ao eixo  $z$ , isto é, que elas suportem a ação da gravidade permanecendo paradas, ao passo que a restrição de estabilidade horizontal (também chamada de estabilidade dinâmica) garante que as caixas não se desloquem com relação

aos eixos  $x$  e  $y$ , isto é, que as caixas resistam a ação da inércia durante o transporte [5].

O trabalho está organizado em seções. A Seção II apresenta a revisão da literatura referente a algumas estratégias que trataram de problemas de carregamento de itens em contêineres bem como das restrições de prioridade e estabilidade. A Seção III apresenta a descrição do PCumC e discute a estratégia proposta para resolvê-lo, incluindo os modelos que a compõe. A Seção IV mostra como modelar as restrições de prioridade, estabilidade vertical e estabilidade horizontal consoante aos modelos adotados. Os experimentos realizados com a estratégia exata e os resultados alcançados são apresentados e discutidos na Seção V. Por fim, na Seção VI são dadas as considerações finais e as perspectivas futuras do trabalho.

## II. REVISÃO DA LITERATURA

As estratégias mais comuns para carregar itens em contêineres utilizaram ideias baseadas em construção de camadas ou blocos de itens. Nelas, os itens são agrupados em camadas ou blocos e depois as camadas ou os blocos são arranjados dentro do contêiner seguindo uma heurística de posicionamento. Dentre essas estratégias é possível mencionar as tratadas em [2], [6], [7], [8] e [9], que utilizaram construção de camadas. Também é possível mencionar as estratégias tratadas em [10], [11] e [12], que utilizaram construção de blocos. Em [7] e [8], houve uma preocupação em diminuir os espaços vazios. Para tanto, [7] pontuou que, minimizando os espaços vazios entre os itens, não há necessidade de preenche-los com espuma ou outros materiais e também contribui para a estabilidade do empacotamento. Em [8], a estratégia consiste em um processo iterativo que seleciona o tipo de caixa para compor a camada e o melhor espaço possível para empacotar a camada formada de caixas do mesmo tipo.

Com relação às estratégias de construção de blocos, em [10] foi proposto construir blocos homogêneos, com itens do mesmo tipo. Segundo [10], blocos homogêneos são mais fáceis de organizar dentro do contêiner e isso corrobora para que o carregamento seja realizado de forma mais rápida. Além disso, salientou-se que blocos formados por itens do mesmo tipo tornam mais difícil o deslizamento de itens nas pilhas, o que de certa forma contribui para a estabilidade do carregamento. Em [12] foram identificadas características presentes nas estratégias que contribuem para a eficácia de heurísticas de posicionamento de caixas. Os autores descobriram, por exemplo, que os tipos de blocos gerados, a representação do espaço livre e a função de avaliação do bloco contribuem significativamente para a eficácia da estratégia de solução. Eles descobriram também que a forma de buscar por blocos (construir os blocos) não impacta significativamente no desempenho da estratégia.

Para encontrar uma solução do PCumC, muitas estratégias associaram heurísticas de construção de blocos ou camadas com meta-heurísticas. Como exemplo, tem-se as estratégias de [6], [8], [13], [14], [15], [16], [17] e [18], que utilizaram algoritmo genético e suas variações. Nessas estratégias com algoritmo genético, a população inicial de cromossomos

consiste em sequências de itens. Para cada cromossomo (isso é, para cada sequência de itens), aplica-se a estratégia de posicionamento adotada (construção de blocos ou camadas) para arranjar os itens da sequência no contêiner. Depois, o empacotamento resultante é avaliado por meio de uma função de aptidão. Em [8], por exemplo, a função de aptidão levou em consideração três aspectos: o volume total carregado, a estabilidade estática e o balanceamento da carga. Outros exemplos são as estratégias de [19], [20] e [21], que utilizaram busca tabu, [9] e [22], que utilizaram recozimento simulado e [23], que utilizaram um algoritmo inspirado em colônia de formigas.

Estratégias que utilizaram busca em árvore foram propostas por [24], [25] e [26]. Em [25], a estratégia de busca em árvore, segundo os próprios autores, pode ser vista como um *branch-and-bound*, em que os nós mais promissores são expandidos e explorados. Em [26] foi utilizada uma árvore ternária, enquanto em [24] foi utilizada uma busca gulosa para percorrer a árvore. Em uma estratégia de busca gulosa, o objetivo é expandir o nó que parece ser o mais promissor com a relação ao objetivo que se deseja alcançar. Assim, em [24], buscou-se escolher uma caixa que gera o melhor aproveitamento do contêiner enquanto houverem caixas que possam ser empacotadas e, assim, tentou-se encontrar uma maneira factível de empacotar a caixa escolhida, respeitando as orientações permitidas para a mesma. Depois de empacotar todas as caixas possíveis, calculou-se o valor de solução e comparou-o com o valor da solução corrente. Se a nova solução fosse melhor do que a corrente, descartava-se a solução anterior e a nova solução passava a ser a corrente.

Alguns métodos exatos foram aplicados para resolver o PCumC. Alguns exemplos podem ser encontrados em [5], [27], [28], [29], [30] e [22]. O algoritmo de [28] foi do tipo *branch-and-bound*. Em [5] foi adotado um *branch-and-cut* para resolver um modelo de programação linear inteira. Em [30] e [22] também há modelos de programação linear inteira resolvidos por um *branch-and-cut*, sendo que [30] enfrentaram problemas relacionados ao alto consumo de memória durante a resolução do modelo. Em [22] foi observado que o tempo requerido para resolver as instâncias cresceu exponencialmente conforme a quantidade de tipos de itens considerados.

Com relação a questão dos métodos exatos demorarem a encontrar uma solução ótima para o problema, em [31] foi apontado que isso se deve pela carência de estratégias para encontrar equivalências no domínio das variáveis e padrões de empacotamento, bem como pela falta de limitantes apertados para o problema. Então, em [31] há a proposta de duas formas de obter limitantes para o PCumC. A primeira forma consiste na *ID bar relaxation* e a segunda consiste na relaxação baseada na construção de fatias (camadas bidimensionais).

A técnica das posições relativas foi utilizada por [32] no PCumC, porém, em um método do tipo *branch-and-prune*. Para executar a poda de soluções inviáveis, os autores utilizaram, dentre outras relaxações, as que foram propostas por [31]. A ideia de posições relativas é saber, dado dois itens  $i$  e  $j$ , se o item  $j$  está empacotado acima ou abaixo, à direita ou

à esquerda, à frente ou à atrás do item  $i$ .

Para tratar a prioridade no carregamento de itens, em [33], [34] e [35] foi proposto empacotar primeiro os itens de alta prioridade. Depois, buscou-se empacotar os itens de prioridade inferior caso houvesse espaço suficiente no contêiner. Outra estratégia para tratar prioridade foi proposta por [9]. Tal estratégia seleciona um subconjunto  $O'$  de itens com alta prioridade. Itens com alta prioridade para os autores são pedidos já expirados, com prazos de entrega mais recentes. Em seguida, verificou-se se todos os itens do subconjunto  $O'$  poderiam ser empacotados sem sobreposição no contêiner. Em caso positivo, selecionou-se um subconjunto  $R'$  de itens de baixa prioridade (pedidos não expirados) e checkou-se se  $R' + O'$  poderiam ser empacotados dentro contêiner. A ideia foi selecionar subconjuntos  $R'$  e  $O'$  para serem testados até encontrar algum que coubesse sem sobreposição no contêiner.

Ainda com relação a restrição prioridade, em [36] há uma formulação com a restrição que só empacota os itens de um nível de prioridade inferior se todos os itens de um nível de prioridade superior puderem ser empacotados. Um problema dessa estratégia é a perda de volume empacotado, pois se nem todos os itens de uma dada prioridade são empacotados no contêiner, a restrição impede que os itens de prioridade inferior sejam arranjados nos espaços restantes. Nota-se ainda que [9], [33], [34] e [35] preocuparam-se com o empacotamento de itens de baixa prioridade caso sobre espaço no contêiner. Porém, tais autores, ao contrário de [36], desenvolveram heurísticas, sendo então mais intuitivo considerar o preenchimento do contêiner com itens de diferentes prioridades.

Com relação à estabilidade vertical, a literatura tem considerado com mais frequência uma condição de fator de suporte. Essa condição impõe que as bases das caixas estejam completamente ou parcialmente suportadas por outras caixas ou pelo chão do contêiner. Estratégias que trataram a estabilidade dessa forma podem ser encontradas em [5], [37] e [38]. Por outro lado, há estratégias em [39] e [8] que resolveram a estabilidade por meio da teoria de equilíbrio estático de corpos rígidos, impondo que o carregamento é estável se as resultantes das forças e de momentos aplicados ao centro de massa de cada caixa são ambas nulas .

### III. ESTRATÉGIA DE RESOLUÇÃO

O PCumC considera um conjunto de caixas  $I$  e um contêiner  $R$ . Assume-se que  $R$  possui largura  $L$ , comprimento  $C$  e altura  $A$ . A quantidade de caixas presente em  $I$  é representada por  $m$ , sendo que cada caixa  $i$  de  $I$  possui largura  $l_i$ , comprimento  $c_i$ , altura  $a_i$  e um valor  $v_i$ . Quando se fala em largura, comprimento e altura, entende-se que essas dimensões estão associadas, respectivamente, aos eixos  $x$ ,  $y$  e  $z$ . O objetivo do problema é determinar quais caixas do conjunto  $I$  devem ser empacotadas no recipiente para que o valor do empacotamento seja máximo. As caixas escolhidas para o carregamento não devem se sobrepor quando colocadas inteiramente dentro do contêiner.

A estratégia exata para encontrar uma solução ótima do PCumC inicia-se com a resolução de um modelo para o

Problema da Mochila Unidimensional (1PM). A solução encontrada para o 1PM consiste em um subconjunto  $S$  de itens de  $I$  tal que o volume total não excede o volume contêiner. No entanto, sobreposições podem ocorrer no empacotamento final, pois, no 1PM, as questões geométricas do problema são desconsideradas. Por isso, faz-se necessário resolver o 3OPP para os itens de  $S$ .

O 3OPP pode ser resolvido diretamente por um modelo de programação por restrições. Porém, considerou-se antes a resolução de um modelo de PI baseado no problema da mochila bidimensional (2PM). A intenção é detectar mais rápido se  $S$  é infactível e evitar chamadas ao modelo de PR. Neste caso, insere-se um corte no 1PM com o intuito de obter uma outra solução, isto é, obter um subconjunto diferente de  $S$ . Por outro lado, caso a solução do modelo associado ao 2PM seja factível, tem-se que o modelo de PR associado ao 3OPP é resolvido para  $S$ . Caso a resposta seja factível, a solução em  $S$  é ótima, caso contrário, insere-se um corte no 1PM para evitar  $S$ .

#### A. Mochila unidimensional

A resolução do problema da mochila unidimensional (1PM), aplicada ao PCumC, busca escolher um subconjunto  $S$  de caixas de valor máximo de tal forma que tais caixas não excedam a capacidade do contêiner, que neste caso, diz respeito ao volume. Além disso, o 1PM não leva em consideração a posição em que cada caixa é empacotada, tampouco busca garantir que não ocorra a sobreposição de caixas. Por não considerar questões de posicionamento e sobreposição, o 1PM é considerado uma relaxação para o PCumC. Logo, o valor ótimo do 1PM consiste em um limitante superior para o valor da solução ótima do PCumC.

Um modelo para o 1PM é apresentado em (1)-(3) e leva em consideração uma variável de decisão binária  $r_i$  que indica se o item  $i$  é escolhido ou não para ser carregado. A função objetivo é dada em (1) e está relacionada à maximização do valor empacotado. A restrição (2) garante que o volume dos itens escolhidos não excederá o volume total do contêiner, ao passo que o domínio da variável de decisão  $r_i$  está em (3), para cada  $i \in I$ .

$$\text{Maximizar } \sum_{i \in I} v_i r_i \quad (1)$$

$$\sum_{i \in I} (l_i c_i a_i) r_i \leq LCA \quad (2)$$

$$r_i \in \{0, 1\}, \quad \forall i \in I \quad (3)$$

#### B. Reduced raster points

Uma vez que um subconjunto  $S$  é determinado, faz-se necessário determinar o domínio de coordenadas  $x$ ,  $y$  e  $z$  possíveis para posicionar as caixas de  $S$ . Uma maneira para fazer isso é discretizar o recipiente em uma malha unitária e considerar todo ponto desta malha para posicionar os itens. Outra maneira consiste em aplicar estratégias para eliminar

pontos redundantes. Nesta caso, tem-se os *reduced raster points* [40].

Em (4), apresenta-se a forma de calcular o conjunto  $X_r$  de coordenadas *reduced raster points* para posicionar os itens no eixo  $x$ . O procedimento para calcular os conjuntos  $Y_r$  e  $Z_r$  é análogo ao mostrado em (4).

$$X_r = \{(L - p)_x \mid p \in X_d\}, \quad (4)$$

em que  $(L - p)_x = \max\{a \in X_d \mid a \leq L - p\}$ . A partir de  $X_r$ , constrói-se o conjunto de coordenadas  $X_i$ , o qual contém todas as coordenadas  $p \in X_r$ , tal que  $p + l_i \leq L$ .

Em (4), o conjunto  $X_d$  refere-se aos *canonical dissections* ou *normal patterns* [41], que é calculado em (5).

$$X_d = \{p \in \mathbb{Z}_+ \mid p = \sum_{i=1}^m \varepsilon_i l_i, 0 \leq p \leq L, \quad (5)$$

$$\varepsilon_i \in \{0, 1\}, i = 1, \dots, m\}$$

### C. Mochila bidimensional

O modelo definido em (6)-(15) recebe  $S$  como entrada e tem por objetivo auxiliar na verificação da viabilidade de se empacotar  $S$  no contêiner. Ele considera as variáveis de decisão binárias  $\beta_{ipq}$ ,  $\gamma_{ipr}$  e  $\delta_{iqu}$ . A variável  $\beta_{ipq}$  assume valor 1 se a caixa  $i$  é empacotada na posição  $(p, q)$  com relação ao plano  $xy$ . A interpretação é análoga para as variáveis  $\gamma_{ipr}$  e  $\delta_{iqu}$ , porém com relação aos planos  $xz$  e  $yz$ , respectivamente.

$$\sum_{i \in S} \sum_{\{p \in X_i \mid s - l_i + 1 \leq p \leq s\}} \sum_{\{q \in Y_i \mid t - c_i + 1 \leq q \leq t\}} a_i \beta_{ipq} \leq A, \quad (6)$$

$$\forall s \in X_r, t \in Y_r$$

$$\sum_{i \in S} \sum_{\{p \in X_i \mid s - l_i + 1 \leq p \leq s\}} \sum_{\{r \in Z_i \mid u - a_i + 1 \leq r \leq u\}} c_i \gamma_{ipr} \leq C, \quad (7)$$

$$\forall s \in X_r, u \in Z_r$$

$$\sum_{i \in S} \sum_{\{q \in Y_i \mid t - c_i + 1 \leq q \leq t\}} \sum_{\{r \in Z_i \mid u - a_i + 1 \leq r \leq u\}} l_i \delta_{iqu} \leq L, \quad (8)$$

$$\forall t \in Y_r, u \in Z_r$$

$$\sum_{t \in Y_i} \beta_{ist} = \sum_{u \in Z_i} \gamma_{isu}, \quad \forall i \in S, \quad \forall s \in X_r \quad (9)$$

$$\sum_{s \in X_i} \beta_{ist} = \sum_{u \in Z_i} \delta_{itu}, \quad \forall i \in S, \quad \forall t \in Y_r \quad (10)$$

$$\sum_{s \in X_i} \gamma_{isu} = \sum_{u \in Z_i} \delta_{itu}, \quad \forall i \in S, \quad \forall u \in Z_r \quad (11)$$

$$\sum_{p \in X_i} \sum_{q \in Y_i} \beta_{ipq} = 1, \quad \forall i \in S \quad (12)$$

$$\beta_{ipq} \in \{0, 1\}, \quad \forall i \in S, \quad \forall p \in X_r, \quad \forall q \in Y_r \quad (13)$$

$$\gamma_{ipr} \in \{0, 1\}, \quad \forall i \in S, \quad \forall p \in X_r, \quad \forall r \in Z_r \quad (14)$$

$$\delta_{iqu} \in \{0, 1\}, \quad \forall i \in S, \quad \forall q \in Y_r, \quad \forall r \in Z_r \quad (15)$$

Na formulação (6)-(15), as restrições em (6) impõem que, para cada reta que é perpendicular a um ponto  $(s, t)$  no plano  $xy$ , a soma das alturas dos itens que interceptam tal reta deve ser menor ou igual a altura do contêiner. As restrições em (7) impõem que, para cada reta que é perpendicular a um ponto  $(s, u)$  no plano  $xz$ , a soma dos comprimentos dos itens que interceptam tal reta deve ser menor ou igual ao comprimento do contêiner. As restrições (8) impõem que, para cada reta que é perpendicular a um ponto  $(t, u)$  no plano  $xz$ , a soma das larguras dos itens que interceptam tal reta deve ser menor ou igual a largura do contêiner. As restrições (9), (10) e (11) garantem que existe uma coordenada  $s$  no eixo  $x$  para empacotar o item  $i$  se e somente se existirem coordenadas  $t$  e  $u$  nos respectivos eixos  $y$  e  $z$  para empacotar  $i$ . As restrições (12) só são satisfeitas caso exista, para todo item  $i$ , uma posição  $(p, q, r)$  dentro do recipiente em que  $i$  possa ser empacotado. Em (13), (14) e (15) são apresentados os domínios das variáveis de decisão.

Duas respostas são possíveis quando o modelo (6)-(15) é resolvido: factível, ou seja, todas as restrições são satisfeitas; ou infactível, ou seja, não existe combinação de pontos que satisfaçam todas as restrições. Se o modelo for infactível, então não é possível empacotar  $S$  e um outro subconjunto precisa ser determinado. Contudo, as questões de sobreposição foram relaxadas no modelo (6)-(15) e, por isso, caso a resposta seja factível para  $S$ , não se pode afirmar que é possível empacotá-lo sem antes checar a sobreposição, isto é, sem antes resolver o 3OPP para  $S$ .

### D. Modelo para o 3OPP

Um modelo básico de programação por restrições para 3OPP consiste em definir a restrição (16) para todo  $i$  e  $j$  do subconjunto de itens  $S$  a ser analisado [32]. Em outras palavras, essa restrição estabelece que se o item  $i$  é empacotado em um ponto  $(X_i, Y_i, Z_i)$ , o item  $j$  (diferente de  $i$ ) deve ser empacotado em um ponto  $(X_j, Y_j, Z_j)$  de tal forma que  $j$  fique acima, abaixo ou dos lados do item  $i$ . Se a restrição for satisfeita para todo par de itens  $i$  e  $j$ , tem-se que o subconjunto  $U$  pode ser empacotado sem sobreposição dentro do contêiner.

$$X_i + l_i \leq X_j \vee X_j + l_j \leq X_i \vee Y_i + c_i \leq Y_j$$

$$\vee Y_j + c_j \leq Y_i \vee Z_i + a_i \leq Z_j \vee Z_j + a_j \leq Z_i, \quad (16)$$

$$(i, j) \in S$$

### E. Cortes

Toda vez que um subconjunto  $S$  não pode ser empacotado no contêiner, o corte representado pela desigualdade (17) precisa ser inserido no modelo para o IPM, que deve então ser resolvido novamente a fim de encontrar um subconjunto

diferente daqueles detectados ineficazes. A soma no lado esquerdo da desigualdade (17) envolve todos os itens de  $S$ . A imposição de que essa soma seja menor ou igual a  $|S| - 1$  garante que um subconjunto diferente seja encontrado da próxima vez que a mochila for resolvida.

$$\sum_{i \in S} r_i \leq |S| - 1 \quad (17)$$

#### IV. RESTRIÇÕES PRÁTICAS

Esta seção apresenta como modelar e incorporar na estratégia exata para o PCumC as restrições de prioridade, estabilidade vertical e estabilidade horizontal.

##### A. Prioridade

A restrição de prioridade é modelada em (18), sendo baseada na proposta de [36] para tratar prioridade no carregamento de itens em aeronaves. Tal restrição considera que os itens são agrupados em níveis de prioridade  $pr = 1, \dots, NPrio$ , em que  $NPrio$  é o número de níveis de prioridade. Itens com prioridade  $pr = 1$  são mais importantes do que itens com prioridade  $pr = 2$  e, assim sucessivamente, até  $pr = NPrio$ . A restrição de prioridade é considerada durante a resolução do 1PM.

$$|Prio_{pr-1}| r_i \leq \sum_{j \in Prio_{pr-1}} r_j, \quad pr = 2, \dots, NPrio, \quad (18)$$

$$\forall i \in Prio_p$$

Em (18),  $Prio_p$  é o conjunto de itens com prioridade  $p$  e  $Prio_{p-1}$  é o conjunto em nível de prioridade anterior, que é mais importante do que o nível  $p$ . Logo, impõe-se que os itens de um nível de prioridade inferior (prioridade  $pr$ ) só poderão ser carregados se todos os itens com um nível de prioridade superior (prioridade  $pr - 1$ ) forem carregados.

##### B. Estabilidade Vertical e Horizontal

As restrições em (19) e (20) são incorporadas ao modelo de PR para o 3OPP para tratar a estabilidade vertical. Elas foram desenvolvidas com base em garantir o suporte da face inferior de cada caixa.

$$\begin{aligned} (Z_j = Z_i + a_i \wedge Z_j - a_i \geq 0 \wedge X_i > X_j - l_i \wedge \\ X_i < X_j + l_j \wedge Y_i > Y_j - c_i \wedge Y_i < Y_j + c_j) \quad (19) \\ \rightarrow \sigma_{ji} = L_{ij}^{[1]} C_{ij}^{[1]}, \quad \forall i, j \in S \end{aligned}$$

em que  $L_{ij}^{[1]} = \min(X_i + l_i, X_j + l_j) - \max(X_i, X_j)$  e  $C_{ij}^{[1]} = \min(Y_i + c_i, Y_j + c_j) - \max(Y_i, Y_j)$ .

$$\sum_{i \in S} \sigma_{ji} \geq \alpha l_j c_j, \quad \forall j \in S \quad (20)$$

As restrições em (19) são do tipo “se-então” e estabelecem, dado um par de caixas  $i, j$ , que se a caixa  $i$  está empacotada imediatamente abaixo e em contato com a caixa  $j$ , a variável auxiliar  $\sigma_{ji}$  assume o valor da área de contato entre essas duas

caixas. Caso a caixa  $i$  não esteja empacotada imediatamente abaixo da caixa  $j$ , a variável auxiliar  $\sigma_{ji}$  assume valor zero. A caixa  $i$  está empacotada imediatamente abaixo da  $j$  se a projeção de suas bases sobre o plano  $xy$  se interceptam e se a coordenada no eixo  $z$  onde a caixa  $i$  termina é igual a coordenada onde a caixa  $j$  começa. Isso é modelado no lado esquerdo de (19), que se refere ao “se”.

A área de contato entre duas caixas  $i$  e  $j$  é calculada pela multiplicação entre  $L_{ij}^{[1]}$  e  $C_{ij}^{[1]}$ . Assim, as restrições (20) estabelecem que no mínimo  $\alpha\%$  da área da base da caixa  $j$  receba suporte, dado pela soma das áreas de contato das caixas que estão empacotadas imediatamente abaixo e contato direto com  $j$ .

As restrições (21) e (22) são incorporadas ao modelo de PR para o 3OPP para tratar a estabilidade horizontal. A variável auxiliar  $\rho_{ji}$  representa o contato entre a face traseira da caixa  $j$  com a face frontal da caixa  $i$ , quando ocorre contato, e a variável  $\tau_{ji}$  representa o contato entre a face lateral direita da caixa  $i$  e a face lateral esquerda da caixa  $j$ . Assim, as restrições em (21) estabelecem que a soma das áreas de todas as caixas que estão empacotadas imediatamente atrás e em contato com a caixa  $j$  seja no mínimo igual a  $\lambda\%$  da área da face traseira da caixa  $j$ . As restrições em (22) estabelecem que a soma das áreas de todas as caixas que estão empacotadas imediatamente ao lado esquerdo e em contato com a caixa  $j$  seja no mínimo  $\mu\%$  da face lateral esquerda da caixa  $j$ . Os contatos  $\rho_{ji}$  e  $\tau_{ji}$  são obtidos de forma análoga a  $\sigma_{ji}$  mostrado em (19).

$$\sum_{i \in S} \rho_{ji} \geq \lambda c_j a_j, \quad \forall j \in S \quad (21)$$

$$\sum_{i \in S} \tau_{ji} \geq \mu l_j a_j, \quad \forall j \in S \quad (22)$$

#### V. EXPERIMENTOS COMPUTACIONAIS

A estratégia exata apresentada neste trabalho foi codificada em linguagem C++. Utilizou-se o pacote comercial de otimização IBM ILOG CPLEX Optimization Studio na versão 12.8 para resolver os modelos presentes na estratégia. O computador usado para realizar os experimentos possui processador Intel Core i5-3570 de 3,4 GHz, 8 GB de memória RAM e sistema Linux Ubuntu 14.04 LTS.

Nos experimentos, buscou-se resolver instâncias do PCumC nas seguintes situações: sem restrições práticas; com as restrições de prioridade; com as restrições de estabilidade vertical; e, com as restrições de estabilidade vertical e horizontal atuando juntas. Utilizou-se 60 instâncias de [5] em cada um dos quatro experimentos. As instâncias escolhidas possuem 10 ou 20 caixas e contêiner de volume  $10^3$ ,  $20^3$  ou  $30^3$ , sendo agrupadas em seis classes. A quantidade de itens e o volume do recipiente que há nas instâncias de cada classe são mostrados na Tabela I. Considerou-se ainda que cada caixa possui demanda unitária, há três níveis de prioridade e  $\alpha$ ,  $\lambda$  e  $\mu$  são iguais a 100%, impondo que as caixas tenham suas bases e faces traseira e lateral esquerda completamente suportadas por outras caixas. Além disso, cada instância teve tempo limite de resolução igual à 3600 segundos.

Tabela I  
INFORMAÇÕES SOBRE AS CLASSES DE INSTÂNCIAS.

Classe	Quantidade de itens	Volume do recipiente
1	10	10 <sup>3</sup>
2	10	20 <sup>3</sup>
3	10	30 <sup>3</sup>
4	20	10 <sup>3</sup>
5	20	20 <sup>3</sup>
6	20	30 <sup>3</sup>

### A. Resultados

As Tabelas II, III, IV e V mostram os resultados obtidos com os experimentos realizados. Tais resultados são com base na média aritmética. Portanto, em cada uma das tabelas referenciadas, a coluna “V.F.O ” mostra o valor médio de função objetivo obtido para cada classe de instâncias, a coluna “Cortes médio ” mostra a quantidade média de cortes inseridos ao longo da resolução, isto é, a quantidade de subconjuntos ótimos para a mochila unidimensional, mas inviáveis do ponto de vista de caberem sem sobreposição no contêiner, e a coluna “Tempo médio ” mostra o tempo médio gasto para cada classe de instâncias.

Os resultados obtidos ao aplicar a abordagem para resolver as instâncias do PCumC sem considerar restrições práticas são mostrados na Tabela II. Nota-se que o tempo requerido para resolver as instâncias foi maior para as classes de instâncias com 20 itens, isto é, para as instâncias das classes 4, 5 e 6. Observou-se também aumento do tempo em função do aumento do tamanho do recipiente nas classes de instâncias. A quantidade de cortes foi maior para as classes de instâncias com 20 itens.

Tabela II  
RESULTADOS OBTIDOS PARA O CASO SEM RESTRIÇÕES PRÁTICAS.

Instâncias	F.V.O médio	Cortes médio	Tempo médio
Classe 1	616,40	6,50	0,28
Classe 2	5841,50	190,60	7,31
Classe 3	18549,90	286,20	15,59
Classe 4	980,10	982,30	3600,00
Classe 5	7996,20	735,10	3600,00
Classe 6	26981,50	701,20	3600,00

A Tabela III mostra os resultados obtidos ao aplicar a estratégia exata considerando a restrição de prioridade. Nota-se que o tempo de resolução foi menor para todas as classes quando comparado com o caso sem restrições práticas. A quantidade média de cortes inseridos foi maior ao se resolver instâncias das classes com 20 itens (classes 4, 5 e 6) do que ao se resolver as instâncias das classes com 10 itens (classes 1, 2 e 3). Além disso, observou-se aumento na quantidade média de cortes em conformidade ao aumento do tamanho do recipiente nas classes com 10 e 20 itens.

Tabela III  
RESULTADOS OBTIDOS PARA O CASO COM PRIORIDADE.

Instâncias	F.V.O médio	Cortes médio	Tempo médio
Classe 1	594,80	1,30	0,21
Classe 2	4520,90	6,70	0,71
Classe 3	12825,50	18,60	0,69
Classe 4	868,80	24,70	1264,89
Classe 5	5577,50	28,80	284,12
Classe 6	16237,50	37,80	16,55

A Tabela IV mostra os resultados para o caso em que se considerou a restrição de estabilidade vertical. Tais resultados demonstram que houve aumento no tempo médio de resolução e na quantidade média de cortes a medida que houve um aumento no número de itens e no tamanho do recipiente.

Tabela IV  
RESULTADOS OBTIDOS PARA O CASO COM ESTABILIDADE VERTICAL.

Instâncias	F.V.O médio	Cortes médio	Tempo médio
Classe 1	610,40	10,10	52,74
Classe 2	5091,60	306,80	51,61
Classe 3	17170,60	370,90	22,72
Classe 4	999,60	667,10	3600,00
Classe 5	8000,00	2024,30	3600,00
Classe 6	51299,20	1852,90	3600,00

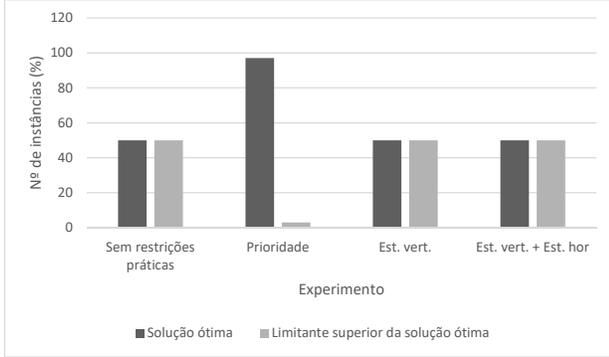
Os resultados obtidos quando se considerou a restrições de estabilidade vertical e horizontal atuando juntas são mostrados na Tabela V. Neste caso, observou-se também aumento no tempo médio de resolução e na quantidade média de cortes em conformidade com o aumento da quantidade de itens e do tamanho do recipiente nas instâncias.

Tabela V  
RESULTADOS OBTIDOS PARA O CASO COM ESTABILIDADE VERTICAL E HORIZONTAL.

Instâncias	F.V.O médio	Cortes médio	Tempo médio
Classe 1	575,80	30,50	328,53
Classe 2	3727,60	622,50	195,58
Classe 3	12536,10	711,80	133,31
Classe 4	999,60	668,40	3600,00
Classe 5	8000,00	2025,30	3600,00
Classe 6	26999,40	1854,80	3600,00

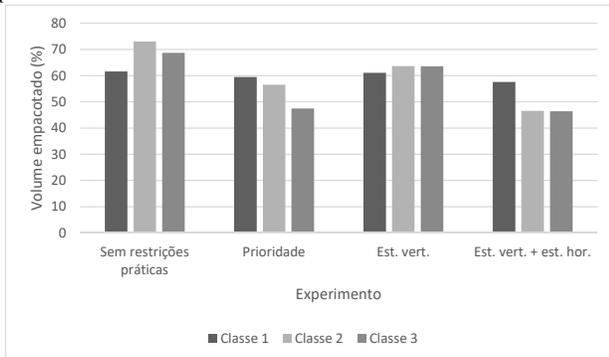
A Figura 1 mostra, para cada experimento, a porcentagem de instâncias para as quais foi possível obter solução ótima e para quais o valor de função objetivo se configura apenas como um limitante superior do valor da solução ótima (casos em que o tempo limite imposto foi atingido). Observa-se que a consideração da restrição de prioridade contribuiu para encontrar a solução ótima para um percentual maior de instâncias quando comparado com os casos sem restrições práticas e com as restrições de estabilidade. Para estes últimos casos, a quantidade de instâncias com solução ótima e do tipo limitante superior foram iguais a 50%, indicando que não houve aumento no número de instâncias com solução ótima ao considerar as restrições de estabilidade, seja, apenas estabilidade vertical, ou estabilidade vertical com horizontal.

Figura 1. Porcentagem de instâncias com solução ótima e limitante superior.



A Figura 2 mostra o volume médio, em porcentagem, que foi empacotado no contêiner em cada experimento para as classes de instâncias 1, 2 e 3. As classes 4, 5 e 6 não foram utilizadas porque, na maioria dos casos, não foi possível encontrar uma solução ótima. Uma vez que a solução não é ótima para o PCumC com ou sem restrições práticas, não há como afirmar que os itens podem ser empacotados. Por isso, para comparar o volume empacotado dado a inserção das restrições práticas, considerou-se somente as classes de instâncias para as quais existe a certeza de que os itens maximizam a função objetivo ao mesmo tempo que podem ser empacotados sem sobreposição no contêiner, o que ocorre para as classes 1, 2 e 3.

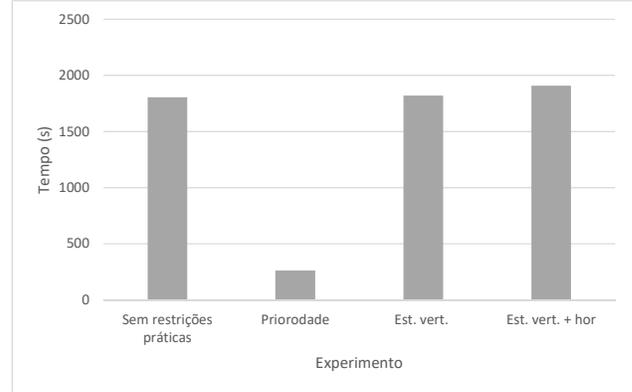
Figura 2. Volume médio (%) empacotado no contêiner em cada tipo de experimento e em cada classe.



A análise da Figura 2 revela que para todas as restrições práticas houve perda de volume empacotado quando comparado com o caso sem restrições práticas. As maiores perdas foram observadas quando se considerou a restrição de prioridade.

Por fim a Figura 3 apresenta o tempo médio, em segundos, requerido para resolver todas as instâncias em cada um dos experimentos. O menor tempo foi observado para o caso em que se considerou prioridade. Para os casos com as restrições de estabilidade vertical e com estabilidade vertical e horizontal, o tempo médio foi alto, ficando entre 1500 e 2000 segundos. Nota-se ainda que o tempo médio de resolução diminuiu dado a inserção da restrição de prioridade, porém aumentou na presença da restrição de estabilidade.

Figura 3. Tempo médio (s) de resolução para cada experimento.



## VI. CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma abordagem para o PCumC composta de três modelos, sendo dois deles de programação inteira e um de programação por restrições. O primeiro modelo de PI não considera restrições para tratar sobreposição, por isso representa uma relaxação para o PCumC. O objetivo por trás da resolução deste modelo relaxado é a obtenção rápida de um limitante superior para a solução ótima do PCumC. Os outros modelos são voltados a representação do subproblema 3OPP. A resolução do 3OPP busca verificar se um conjunto de itens tridimensionais pode ou não ser empacotado dentro de um contêiner. Além disso, mostrou-se como modelar restrições práticas dentro da estratégia desenvolvida. Essas restrições são para garantir o carregamento de itens de alta prioridade (restrição de prioridade) e que os itens terão suas bases e faces laterais esquerda e traseira completamente suportadas por outras caixas ou pelo chão e laterais do contêiner (restrições de estabilidade vertical e horizontal).

A estratégia exata foi aplicada sobre 60 instâncias considerando diferentes cenários. Primeiro, resolveu-se as instâncias sem considerar restrições práticas. Depois, resolveu-se as instâncias considerando cada uma das três restrições práticas tratadas no trabalho. Buscou-se com os experimentos computacionais verificar o impacto da inserção das restrições práticas no que diz respeito ao volume empacotado e tempo de resolução. Com relação ao volume empacotado, observou-se a sua diminuição com a inserção das restrições práticas. As maiores perdas de volume foram observadas quando se considerou a restrição de prioridade. Com relação ao tempo de resolução, observou-se uma diminuição de 85,52% com a inserção da restrição de prioridade e um aumento de 0,96% com a inserção da restrição de estabilidade vertical e 5,86% com a inserção das restrições de estabilidade vertical e horizontal.

Pretende-se, como perspectiva futura deste trabalho, trabalhar com outras restrições práticas relacionadas à problemas de empacotamento de itens em contêineres. Melhorias com relação ao desempenho da abordagem podem ser obtidas investigando novos limitantes para o problema. Explorar a técnica de programação por restrições a fim de alcançar um

desempenho maior com relação a redução do domínio das variáveis de decisão pode ser outro caminho de trabalho futuro.

#### 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).

#### REFERENCES

- [1] A. Bortfeldt and G. Wäscher, "Constraints in container loading—a state-of-the-art review," *European Journal of Operational Research*, vol. 229, no. 1, pp. 1–20, 2013.
- [2] E. E. Bischoff and M. Ratcliff, "Issues in the development of approaches to container loading," *Omega*, vol. 23, no. 4, pp. 377–390, 1995.
- [3] A. Caprara and M. Monaci, "On the two-dimensional knapsack problem," *Operations Research Letters*, vol. 32, no. 1, pp. 5–14, 2004.
- [4] T. A. Queiroz, P. H. D. B. Hokama, R. C. S. Schouery, and F. K. Miyazawa, "Two-dimensional disjunctively constrained knapsack problem: Heuristic and exact approaches," *Computers & Industrial Engineering*, vol. 105, pp. 313–328, 2017.
- [5] L. Junqueira, R. Morabito, and D. S. Yamashita, "Three-dimensional container loading models with cargo stability and load bearing constraints," *Computers & Operations Research*, vol. 39, no. 1, pp. 74–85, 2012.
- [6] A. Bortfeldt and H. Gehring, "A hybrid genetic algorithm for the container loading problem," *European Journal of Operational Research*, vol. 131, no. 1, pp. 143–161, 2001.
- [7] D. Pisinger, "Heuristics for the container loading problem," *European Journal of Operational Research*, vol. 141, no. 2, pp. 382–392, 2002.
- [8] A. G. Ramos, J. F. Oliveira, J. F. Gonçalves, and M. P. Lopes, "A container loading algorithm with static mechanical equilibrium stability constraints," *Transportation Research Part B: Methodological*, vol. 91, pp. 565–581, 2016.
- [9] L. Sheng, S. Xiuqin, C. Changjian, Z. Hongxia, S. Dayong, and W. Feiyue, "Heuristic algorithm for the container loading problem with multiple constraints," *Computers & Industrial Engineering*, vol. 108, pp. 149–164, 2017.
- [10] M. Eley, "Solving container loading problems by block arrangement," *European Journal of Operational Research*, vol. 141, no. 2, pp. 393–409, 2002.
- [11] D. Zhang, Y. Peng, and S. C. Leung, "A heuristic block-loading algorithm based on multi-layer search for the container loading problem," *Computers & Operations Research*, vol. 39, no. 10, pp. 2267–2276, 2012.
- [12] W. Zhu, W.-C. Oon, A. Lim, and Y. Weng, "The six elements to block-building approaches for the single container loading problem," *Applied Intelligence*, vol. 37, no. 3, pp. 431–445, 2012.
- [13] H. Gehring and A. Bortfeldt, "A genetic algorithm for solving the container loading problem," *International Transactions in Operational Research*, vol. 4, no. 5-6, pp. 401–418, 1997.
- [14] L. J. P. de Araújo and P. Pinheiro, "Combining heuristics backtracking and genetic algorithm to solve the container loading problem with weight distribution," in *Soft Computing Models in Industrial and Environmental Applications, 5th International Workshop (SOCO 2010)*. Springer, 2010, pp. 95–102.
- [15] J. F. Gonçalves and M. G. Resende, "A parallel multi-population biased random-key genetic algorithm for a container loading problem," *Computers & Operations Research*, vol. 39, no. 2, pp. 179–190, 2012.
- [16] J.-N. Zheng, C.-F. Chien, and M. Gen, "Multi-objective multi-population biased random-key genetic algorithm for the 3-d container loading problem," *Computers & Industrial Engineering*, vol. 89, pp. 80–87, 2015.
- [17] Y.-H. Huang, F. Hwang, and H.-C. Lu, "An effective placement method for the single container loading problem," *Computers & Industrial Engineering*, vol. 97, pp. 212–221, 2016.
- [18] T. Jamrus and C.-F. Chien, "Extended priority-based hybrid genetic algorithm for the less-than-container loading problem," *Computers & Industrial Engineering*, vol. 96, pp. 227–236, 2016.
- [19] A. Bortfeldt, H. Gehring, and D. Mack, "A parallel tabu search algorithm for solving the container loading problem," *Parallel Computing*, vol. 29, no. 5, pp. 641–662, 2003.
- [20] J. Liu, Y. Yue, Z. Dong, C. Maple, and M. Keech, "A novel hybrid tabu search approach to container loading," *Computers & Operations Research*, vol. 38, no. 4, pp. 797–807, 2011.
- [21] Y. Tao and F. Wang, "An effective tabu search approach with improved loading algorithms for the 3l-cvrp," *Computers & Operations Research*, vol. 55, pp. 127–140, 2015.
- [22] H. Mostaghimi Ghomi, B. G. St Amour, and W. Abdul-Kader, "Three-dimensional container loading: A simulated annealing approach," *International Journal of Applied Engineering Research*, vol. 12, no. 7, p. 1290, 2017.
- [23] M. Hifi, S. Saleh, and L. Wu, "A hybrid guided neighborhood search for the disjunctively constrained knapsack problem," *Cogent Engineering*, vol. 2, no. 1, p. 1068969, 2015.
- [24] S. G. Christensen and D. M. Rousøe, "Container loading with multi-drop constraints," *International Transactions in Operational Research*, vol. 16, no. 6, pp. 727–743, 2009.
- [25] I. Araya and M.-C. Riff, "A beam search approach to the container loading problem," *Computers & Operations Research*, vol. 43, pp. 100–107, 2014.
- [26] L. Sheng, Z. Hongxia, D. Xisong, and C. Changjian, "A heuristic algorithm for container loading of pallets with infill boxes," *European Journal of Operational Research*, vol. 252, no. 3, pp. 728–736, 2016.
- [27] G. Scheithauer, "Algorithms for the container loading problem," in *Operations Research Proceedings 1991*. Springer, 1992, pp. 445–452.
- [28] S. Martello, D. Pisinger, and D. Vigo, "The three-dimensional bin packing problem," *Operations Research*, vol. 48, no. 2, pp. 256–267, 2000.
- [29] M. Hifi, "Exact algorithms for unconstrained three-dimensional cutting problems: a comparative study," *Computers & Operations Research*, vol. 31, no. 5, pp. 657–674, 2004.
- [30] R. N. Cardoso, B. L. Pereira, J. Fonseca, M. Ferreira, and J. Tavares, "Automated planning integrated with linear programming applied in the container loading problem," *IFAC Proceedings Volumes*, vol. 46, no. 24, pp. 153–158, 2013.
- [31] G. Scheithauer, "Lp-based bounds for the container and multi-container loading problem," *International Transactions in Operational Research*, vol. 6, no. 2, pp. 199–213, 1999.
- [32] M. Mesyagutov, G. Scheithauer, and G. Belov, "New constraint programming approaches for 3d orthogonal packing," *Technische Universität Dresden, Dresden, Germany*, 2012.
- [33] J. Ren, Y. Tian, and T. Sawaragi, "A tree search method for the container loading problem with shipment priority," *European Journal of Operational Research*, vol. 214, no. 3, pp. 526–535, 2011.
- [34] N. Wang, A. Lim, and W. Zhu, "A multi-round partial beam search approach for the single container loading problem with shipment priority," *International Journal of Production Economics*, vol. 145, no. 2, pp. 531–540, 2013.
- [35] T. Tian, W. Zhu, A. Lim, and L. Wei, "The multiple container loading problem with preference," *European Journal of Operational Research*, vol. 248, no. 1, pp. 84–94, 2016.
- [36] W. Vancroonenburg, J. Verstichel, K. Tavernier, and G. V. Berghe, "Automatic air cargo selection and weight balancing: a mixed integer programming approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 65, pp. 70–83, 2014.
- [37] S. Ceschia, A. Schaefer, and T. Stützle, "Local search techniques for a routing-packing problem," *Computers & industrial engineering*, vol. 66, no. 4, pp. 1138–1149, 2013.
- [38] A. Bortfeldt, T. Hahn, D. Männel, and L. Mönch, "Hybrid algorithms for the vehicle routing problem with clustered backhauls and 3d loading constraints," *European Journal of Operational Research*, vol. 243, no. 1, pp. 82–96, 2015.
- [39] T. A. Queiroz and F. K. Miyazawa, "Order and static stability into the strip packing problem," *Annals of Operations Research*, vol. 223, no. 1, pp. 137–154, 2014.
- [40] G. Scheithauer and J. Terno, "The g4-heuristic for the pallet loading problem," *Journal of the Operational Research Society*, vol. 47, no. 4, pp. 511–522, 1996.
- [41] J. Herz, "Recursive computational procedure for two-dimensional stock cutting," *IBM Journal of Research and Development*, vol. 16, no. 5, pp. 462–469, 1972.