



UNIVERSIDADE
ESTADUAL DE LONDRINA

BERNARDO ABREU DA CRUZ

**PROBLEMA DE CARREGAMENTO DE VEÍCULO
MULTICOMPARTIMENTADO**

Londrina
Novembro / 2023

BERNARDO ABREU DA CRUZ

**PROBLEMA DE CARREGAMENTO DE VEÍCULO
MULTICOMPARTIMENTADO**

Dissertação de mestrado apresentada ao Departamento de Matemática da Universidade Estadual de Londrina, como requisito parcial para a obtenção do Título de MESTRE em Matemática Aplicada e Computacional.

Orientador: Prof^a. Dr^a. Aline Aparecida de
Souza Leão

Londrina
Novembro / 2023

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Cruz, Bernardo Abreu da.

Problema de Carregamento de Veículo Multicompartimentado / Bernardo Abreu da Cruz. - Londrina, 2023.
95 f.

Orientador: Aline Aparecida de Souza Leão.

Dissertação (Mestrado em Matemática Aplicada e Computacional) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Matemática Aplicada e Computacional, 2023.
Inclui bibliografia.

1. Otimização - Tese. 2. Empacotamento Tridimensional - Tese. 3. Heurística Construtiva - Tese. 4. Contêiner multicompartimentado - Tese. I. Aparecida de Souza Leão, Aline. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Matemática Aplicada e Computacional. III. Título.

CDU 51

BERNARDO ABREU DA CRUZ

PROBLEMA DE CARREGAMENTO DE VEÍCULO MULTICOMPARTIMENTADO

Dissertação de mestrado apresentada ao Departamento de Matemática da Universidade Estadual de Londrina, como requisito parcial para a obtenção do Título de MESTRE em Matemática Aplicada e Computacional.

BANCA EXAMINADORA

Prof^a. Dr^a. Aline Aparecida de Souza Leão
Universidade Estadual de Londrina

Prof. Dr. Marcos Mansano Furlan
Universidade Federal da Grande Dourados

Prof. Dr. Pedro Belin Castellucci
Universidade Federal de Santa Catarina

Londrina, 28 de Novembro de 2023.

Aos meus pais, Nilvia e José Carlos.

AGRADECIMENTOS

Meu agradecimento a todos que, de alguma forma, contribuíram para concretização deste trabalho e, de uma maneira especial, agradeço:

- A minha mãe Nilvia e ao meu pai José Carlos, pelo incentivo, apoio incondicional, amor e dedicação. Espero que um dia consiga retribuir tudo o que já fizeram por mim.
- A minha família, que são a base de tudo e por sempre estarem ao meu lado, me incentivando e apoiando em todos os momentos.
- Aos professores do PGMAC, pela disposição e todo conhecimento compartilhado, e em especial, a minha orientadora, professora Dr.^a Aline Leão, por todas as instruções, dedicação, compreensão e paciência comigo em todos os momentos.
- Aos professores integrantes das bancas de qualificação e defesa, pela disponibilidade e contribuições para a melhoria deste trabalho.
- Aos meus colegas do mestrado, pelos momentos de alegria compartilhados na salinha do PGMAC e pelo companheirismo nas dificuldades durante todo este período. Em especial, agradeço a amiga e colega Mariana Nunes, que esteve comigo desde o início do mestrado e por ter me acolhido quando cheguei em Londrina.
- Aos meus amigos, de todos os lugares, pela amizade e incentivo. De maneira especial, agradeço à Patrícia Stülp e Cristian Fagundes, pela amizade, apoio e disponibilidade para ajudar em todos os momentos que precisei. São pessoas muito especiais e amigos que admiro muito.
- À Deus, aos mentores e amigos do plano espiritual, por esta vida e por mais uma oportunidade de estar aqui evoluindo. Sei que mesmo sem vê-los, sempre estão me acompanhando.
- À Fundação Araucária, pelo apoio financeiro, convênio 024/2021.

*É justo que muito custe aquilo que muito vale.
(Santa Teresa d'Ávila)*

CRUZ, Bernardo Abreu da. **Problema de Carregamento de Veículo Multicompartimentado** 2023. 65 f. Dissertação (Mestrado em Matemática Aplicada e Computacional) – Universidade Estadual de Londrina, Londrina, ano.

RESUMO

O problema de carregamento de contêiner multicompartimentado estudado neste trabalho consiste em empacotar itens em veículos caracterizados por carrocerias divididas em compartimentos. Este problema surge no empacotamento e distribuição de produtos da indústria de bebidas, em que o veículo deve seguir uma rota predefinida e atender a demanda em sua totalidade. Os itens são acomodados ortogonalmente em camadas, respeitando as restrições práticas: orientação dos itens, estabilidade vertical e resistência de empilhamento, que devem ser satisfeitas durante toda atividade. O objetivo é minimizar o manuseio das caixas ao longo do percurso de entrega e os desvios de balanceamento de peso. Como na literatura, este problema foi resolvido por uma matheurística, em que alguns modelos de programação inteira mista resolvidos podem demandar um alto tempo computacional, o objetivo deste trabalho é desenvolver uma heurística construtiva com o propósito de obter soluções de boa qualidade e em menor tempo computacional. A heurística consiste em distribuir os itens em camadas completas e incompletas. As camadas completas são compostas por um mesmo tipo de item, acomodados horizontalmente, sem empilhamento e com a quantidade máxima de caixas. O número máximo de itens em uma camada completa é definido durante a fase de pré-processamento e durante a heurística elas são carregadas nos compartimentos em pilha. As camadas incompletas são compostas por mais de um tipo de item. As caixas são acomodadas em pilhas verticais e devem ocupar as posições mais elevadas dos compartimentos, acima das camadas completas. Os experimentos computacionais realizados mostraram que a utilização da heurística proposta possibilitou alcançar soluções factíveis e em um reduzido tempo computacional, obtendo soluções 0.0033 segundos. Os resultados obtidos para os custos de remanejamento e os desvios de balanceamento apresentaram valores satisfatórios. Os valores alcançados mostraram-se condizentes com os números obtidos com resoluções presentes literatura em termos de custos de remanejamento e de desvios de balanceamento.

Palavras-chave: Otimização. Empacotamento tridimensional. Contêiner multicompartimentado. Heurística construtiva.

CRUZ, Bernardo Abreu da. **Multicompartment Vehicle Loading Problem**. 2023. 65 f. Dissertação (Mestrado em Matemática Aplicada e Computacional) – Universidade Estadual de Londrina, Londrina, ano.

ABSTRACT

The multi-compartment container loading problem addressed in this study consists of packing a set of boxes in vehicles divided into compartments. This problem can be found in the packing and distribution of beverage industries, where a vehicle must follow a predefined route and meet the total demand. The boxes must be packed in orthogonal layers satisfying the following hard constraints during the route: box orientation, vertical stability and load-bearing strength of the boxes. The objective is to minimize the handling operations of the boxes during the delivery route and the load balancing deviations. Since this problem has been solved in the literature by matheuristic, where the solution of some mixed integer linear programming models can take a high computational time, we propose a constructive heuristic in order to obtain good solutions in a shorter computational time. The heuristic consists of packing the boxes in completed and uncompleted layers. A completed layer comprises the maximum number of boxes allowed of the same type and without stacking. The maximum number of boxes in each completed layer is defined in a preprocessing phase, and during the heuristic method, they are stacked in the compartments. An uncompleted layer is composed of different types of boxes that are packed following a wall-building procedure. In addition, it must be packed on the last completed layer in a compartment. Computational experiments show that the proposed heuristic can find good feasible solutions in a short computational time, determining solutions within 0.003 seconds on average. The results obtained for the handling operations and the load balancing deviations were satisfactory. The values showed to be consistent with the solutions described in the literature.

Keywords: Optimization. Three-dimensional packaging. Multi-compartmented container. Constructive heuristic.

LISTA DE FIGURAS

2.1	Problema de corte unidimensional.	22
2.2	Problema de corte bidimensional.	22
2.3	Problema de corte tridimensional.	23
2.4	Tipologia para MCVRP.	26
3.1	Veículo multicompartimentado.	31
3.2	Item de referência com a nomenclatura de cada face.	32
3.3	Organização dos itens em camadas.	32
3.4	Camadas acomodadas em um compartimento.	33
3.5	Sistema de coordenadas para um contêiner com oito compartimentos.	36
3.6	Sistema de coordenadas para um compartimento.	36
3.7	Exemplo de possíveis orientações para o empacotamento de um item.	36
3.8	Posições para carregamento dos itens nos compartimentos.	37
3.9	Ilustração de itens acomodados em camadas completas e incompletas.	39
3.10	Estabilidade para o caso de um item apoiado em outro item.	42
3.11	Estabilidade para o caso de um item apoiado em uma camada.	42
3.12	Estabilidade para o caso de uma camada apoiada em outra camada.	43
3.13	Exemplo de estabilidade horizontal no eixo x	45
3.14	Exemplo de estabilidade horizontal no eixo y	45
3.15	Empilhamento de itens de diferentes destinos.	47
3.16	Centros de gravidade para cálculo dos desvios de balanceamento.	49
3.17	Carregamento de itens sem remanejamento necessário.	52
3.18	Carregamento de itens com remanejamento necessário.	53
4.1	Exemplos de camadas completas.	56
4.2	Exemplo de uma camada incompleta.	56
4.3	Exemplo de distribuição dos itens em pilhas.	57
4.4	Exemplo de padrão de empacotamento em um compartimento.	57
4.5	Exemplo de espaço destinado a uma camada incompleta.	58
4.6	Exemplo de ordem de carregamento para um veículo com 8 compartimentos.	58
4.7	Exemplo de parede.	67
4.8	Exemplos de pilha e blocos.	68
4.9	Exemplo dos espaços secundários em uma parede.	68
4.10	Exemplo de um segundo nível de espaços secundários em uma parede.	69
4.11	Algoritmos da Heurística.	75

LISTA DE TABELAS

5.1	Informações do contêiner multicompartimentado.	77
5.2	Tipos de compartimentos e suas dimensões.	77
5.3	Dados dos itens utilizados nos experimentos computacionais.	78
5.5	Demandas dos tipos de itens por paradas.	79
5.6	Sequências de paradas consideradas nas instâncias.	80
5.7	Informações da resolução dos exemplares 1I a 10I.	81

LISTA DE SÍMBOLOS E ABREVIações

b_i	Quantidade disponível (ou a demanda) do item de tipo $i \in M$.
C	Conjunto dos compartimentos que compõem o contêiner.
cR	Valor total das penalidades de remanejamento dos itens.
cB	Valor total das penalidades dos desvios de balanceamento.
d_{ik}	Demanda do item de tipo $i \in M$ no destino $k \in K$.
esp^{livre}	Dimensões do espaço livre para acomodar itens em uma camada incompleta, em que $esp^{livre} = (esp^{livre}(x), esp^{livre}(y), esp^{livre}(z))$.
esp^{altura}	Dimensões do espaço secundário altura (acima de uma parede), em que $esp^{livre} = (esp^{altura}(x), esp^{altura}(y), esp^{altura}(z))$.
$esp^{alturaInc}$	Dimensões do espaço secundário altura incompleta de uma parede, em que $esp^{livre} = (esp^{alturaInc}(x), esp^{alturaInc}(y), esp^{alturaInc}(z))$.
$esp^{lateral}$	Dimensões do espaço secundário lateral de uma parede, em que $esp^{livre} = (esp^{lateral}(x), esp^{lateral}(y), esp^{lateral}(z))$.
$esp^{frontal}$	Dimensões do espaço secundário frontal de uma parede, em que $esp^{livre} = (esp^{frontal}(x), esp^{frontal}(y), esp^{frontal}(z))$.
g^{vx}, g^{vy}	Centro de gravidade ideal para os itens acomodados no contêiner em relação aos eixos x e y , definido a partir do sistema de coordenadas do contêiner.
g_k^x, g_k^y	Variáveis auxiliares que definem o centro de gravidade real dos itens demandados pela parada k , com $k' \geq k$, em relação aos eixos x e y , respectivamente, definido a partir do sistemas de coordenadas do contêiner.
K	Conjunto de destinos das entregas.
l_i, w_i, h_i	Comprimento (eixo x), largura (eixo y) e altura (eixo z), respectivamente, de um item i .
L, W, H	Comprimento (eixo x), largura (eixo y) e altura (eixo z), respectivamente, do contêiner.
L^{comp}, W^{comp}	Comprimento (eixo x) e largura (eixo y), respectivamente, dos compartimentos.
$list^{camadas}$	Lista de camadas completas que preenchem totalmente um compartimento.
$list^{verificados}$	Lista dos tipos de itens que já foram testados no empacotamento.
$list^{paredes}$	Lista de paredes que compõem uma camada incompleta.
$list_c^{carga}$	Lista de empacotamento dos itens $i \in M'$ no compartimento $c \in C$.

h_c :	Limitante superior, relativo ao eixo z , para o espaço livre destinado as camadas completas acomodadas no compartimento $c \in C$.
H_c^{comp}	Altura (eixo z) do compartimento $c \in C$.
M	Conjunto de todos os itens demandados.
\bar{M}	Conjunto de itens residuais.
\mathcal{M}	Número suficientemente grande.
N	Conjunto de todas as camadas geradas, i.e., $N = N' \cup N''$.
N'	Conjunto das camadas completas.
N''	Conjunto das camadas incompletas.
$nivel$	Variável de verificação de nível das paredes, em que $nivel = 0$, o empacotamento de itens ocorre na base da camada incompleta, caso contrário $nivel = 1$.
p, p', s	Coordenadas em relação ao eixo x .
p_c, q_c	Posições nos eixos x e y , respectivamente, que o compartimento $c \in C$ está posicionado no contêiner.
P_i	Peso de um item do item tipo i .
P_{jk}	Peso da camadas $j \in N$ quando apenas os itens das paradas $k' \geq k$ estão carregados.
P_k^Δ	Peso total da carga quando apenas os itens destinados a parada k e posteriores ($k' \geq k$) estão carregados no contêiner.
$pilha^{inc}$	Variável de verificação da construção de pilhas incompletas, $pilha^{inc} = 1$ caso tenha gerado pilha incompleta, caso contrário $pilha^{inc} = 0$ $tamanho^{parede}$.
qR	quantidade total de itens remanejados.
q, q', t	Coordenadas em relação ao eixo y .
Q_i^{max}	Quantidade máxima de itens do tipo i que podem ser acomodados em uma camada.
Q_{ik}^{mistas}	Conjunto com a quantidade de itens i da parada k alocados em uma camada completa mista.
$Q_{itensPilha}$	Conjunto que armazena a quantidade de itens nas pilha.
Q_{pilhas}	Conjunto que armazena a quantidade de pilhas nas paredes.
r, r', u	Coordenadas em relação ao eixo z .
T_c	Tempo computacional para gerar as camadas completas.
T_i	Tempo computacional para gerar as camadas incompletas.
$tamanho^{bloco}$	Dimensões do espaço ocupado pelos itens $i \in M'$ acomodados na parte completa de um bloco, em que $tamanho^{bloco} = (tamanho^{bloco}(x), tamanho^{bloco}(y), tamanho^{bloco}(z))$.

$tamanho^{blocoInc}$	Dimensões do espaço ocupado pelos itens $i \in M'$ acomodados na parte incompleta de um bloco, em que $tamanho^{blocoInc} = (tamanho^{blocoInc}(x), tamanho^{blocoInc}(y), tamanho^{blocoInc}(z))$.
x_{ipq}	A variável de decisão que um item, com orientação $i = 1, 2$, está acomodado no palete quando o canto inferior esquerdo do item ocupa a posição (p, q) no sistema de coordenadas estabelecido para o palete.
$x_{jkpqr c}$	Variável binária de decisão que é igual a 1 se o item do tipo $i \in \overline{M}$ e do destino $k \in K$, é colocado na posição (p, q, r) , com $p \in X_i^0$, $q \in Y_i^0$ e $r \in Z_{ic}^0$, do compartimento $c \in C$; e caso contrário, é igual a 0.
\hat{x}_{ikrc}	Variável que descreve a quantidade de itens do destino $k \in K$ em uma camada do tipo $i \in M$ colocada na posição $(0, 0, r)$, com $r \in Z_{ic}^0$, no compartimento $c \in C$.
\hat{x}_{irc}^a	Variável binária de decisão que é igual a 1 se há itens em uma camada do tipo $i \in M$, colocada na posição $(0, 0, r)$, com $r \in Z_{ic}^0$, no compartimento $c \in C$; e caso contrário, é igual a 0.
\hat{x}_{krc}^b	Variável binária de decisão para verificar se uma camada foi alocada na posição $(0, 0, r)$, com $r \in Z_c^0$, no compartimento $c \in C$ e com itens demandados na parada $k \in K$, então é igual a 1; caso contrário, é igual a 0.
\hat{x}_{krc}^d	Variável binária de decisão que determina se uma camada completa alocada na posição $(0, 0, r)$, $r \in Z_{ic}^0$, no compartimento $c \in C$ contém itens da parada $k \in K$, então é igual a 1; caso contrário, é igual a 0.
u_{ipqrs}	Parâmetro que, quando uma caixa está posicionada em (p, q) , as posições internas (r, s) dessa caixa estejam indisponíveis para acomodar qualquer outro item.
z_{jkc}	Variável inteira de decisão que determina a quantidade de itens de um mesmo tipo j , acomodados em camadas completas ($j \in M$) e em camadas incompletas ($j \in \overline{M}$), a serem remanejados no compartimento $c \in C$ na parada $k \in K$.
$\%res$	Porcentagem de itens residuais em relação a demanda total.
α	Parâmetro de estabilidade vertical $\alpha \in \mathbb{R}$, tal que $0 \leq \alpha \leq 1$. Para $\alpha = 1$, é necessário que a face inferior de qualquer item esteja 100% apoiada em outros itens, camadas ou na base do compartimento. Para $\alpha = 0$, indica-se que não há exigência de estabilidade vertical para os itens.

β	Parâmetro de estabilidade horizontal para o eixo x $\beta \in \mathbb{R}$, tal que $0 \leq \beta \leq 1$. Para $\beta = 1$, é necessário que a face lateral de qualquer item esteja 100% apoiada em outros itens, ou na lateral do compartimento. Para $\beta = 0$, indica-se que não há exigência de estabilidade horizontal no eixo x para os itens.
γ	Parâmetro de estabilidade horizontal para o eixo y $\gamma \in \mathbb{R}$, tal que $0 \leq \gamma \leq 1$. Para $\gamma = 1$, é necessário que a face frontal de qualquer item esteja 100% apoiada em outros itens, ou no fundo do compartimento. Para $\gamma = 0$, indica-se que não há exigência de estabilidade horizontal no eixo y para os itens.
$\varepsilon^x, \varepsilon^y$	Coeficientes de tolerância para os centros de gravidades, utilizados no calculo das variáveis θ_k^x e θ_k^y , com $k \in K$ e sendo $\varepsilon^x \geq 0$ e $\varepsilon^y \geq 0$.
θ_k^x, θ_k^y	Variáveis de decisão que determinam os desvios de balanceamento de peso no contêiner em relação aos eixos x e y , respectivamente, dos itens demandados pela para k , com $k' \geq k$.
σ_i	Pressão máxima suportada em qualquer ponto da face superior de um item do tipo i .
τ_i^x, τ_i^y	Centro de gravidade do item $i \in \overline{M}$ em relação aos eixos x e y , respectivamente.
ψ_i	Quantidade mínima de itens que uma camada completa do tipo i precisa ter para garantir estabilidade vertical para os itens, ou camadas, acomodados em posições acima dela.
ω^{dev}	Peso relativo aos desvios de balanceamento de peso no contêiner.
ω^z	Peso relativo para o remanejamento de itens.
ω^{zq}, ω^{zp}	Pesos relativos para a quantidade e o peso dos itens remanejados, respectivamente.

SUMÁRIO

1	INTRODUÇÃO	17
2	PROBLEMAS DE CORTE E EMPACOTAMENTO: CLASSIFICAÇÃO E REVISÃO BIBLIOGRÁFICA	20
2.1	TIPOS DE PROBLEMAS	20
2.2	PROBLEMAS DE CORTE E EMPACOTAMENTO	21
2.3	PROBLEMAS DE CARREGAMENTO DE VEÍCULOS MULTICOMPARTIMENTADOS	24
3	APLICAÇÃO ESTUDADA	31
3.1	DEFINIÇÃO DO PROBLEMA	31
3.2	MODELAGEM MATEMÁTICA	34
3.2.1	Posições para Empacotamento	34
3.2.2	Restrições Básicas e de Demanda	39
3.2.3	Restrições de Estabilidade	41
3.2.4	Restrições de Resistência ao Empilhamento	46
3.2.5	Balanceamento de Peso	49
3.2.6	Remanejamento de Itens	52
4	MÉTODO DE SOLUÇÃO PROPOSTO	55
4.1	CONSTRUÇÃO DAS CAMADAS COMPLETAS	60
4.2	CONSTRUÇÃO DAS CAMADAS INCOMPLETAS	67
5	EXPERIMENTOS COMPUTACIONAIS	76
5.1	INSTÂNCIAS	76
5.2	RESULTADOS	80
6	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	83
	REFERÊNCIAS	85

1 INTRODUÇÃO

Em diversos ramos da indústria, o processo de carregamento é uma importante etapa do transporte de mercadorias, principalmente para aqueles setores que operam com a logística de entrega e/ou coleta de produtos. Estas atividades estão fortemente relacionadas ao lucro de uma empresa e por isso, é relevante que sejam desenvolvidas de maneira eficiente. O transporte de produtos utilizando contêineres é frequentemente observado, por serem objetos versáteis para diferentes modais, permitindo o empacotamento de diferentes tipos de itens. Conforme Ranck Jr. et al. (2019), pode-se associar os custos desta atividade às rotas adotadas na distribuição, ao planejamento e dimensionamento utilizado para a construção dos lotes e no processo de carregamento dos produtos. A redução de custo no processo de transporte é um tópico relevante na área de logística, pois equivale a mais de 10% do PIB nacional, em que os valores com transporte correspondem a mais de 50% deste percentual (RANCK Jr., 2014).

Problemas de carregamento podem ser descritos pelo empacotamento de itens menores, aqui considerados como itens ou caixas, em unidades maiores, que vamos identificar como objetos ou contêineres. Pelo ponto de vista matemático, problemas de corte que descrevem situações em que se precisa cortar itens menores de objetos, como chapas ou bobinas, podem ser analisados de maneira equivalente. Estes problemas são considerados na literatura como Problemas de Corte e Empacotamento (PCE). Os PCE são amplamente estudados, pois permitem descrever diferentes situações e, com isso, Wäscher, Haußner e Schumann (2007) propuseram uma tipologia, a fim de estabelecer uma categorização destes tipos de problemas que estivesse mais condizente com as situações encontradas na prática.

Para a classificação dos PCE, alguns critérios foram utilizados e dentre eles, a dimensionalidade ganha uma relevância, sendo importante para descrever a geometria dos padrões de corte e empacotamento. Neste trabalho, realizamos a revisão de alguns estudos acerca de problemas unidimensionais, bidimensionais e tridimensionais. Visando o caso tridimensional, Bortfeldt e Wäscher (2013) identificam uma série de restrições, que em conjunto com as restrições geométricas básicas, contribuem para uma melhor descrição de problemas reais.

Nos problemas de carregamento de contêineres, a utilização de contêineres multicompartimentados apresenta benefícios interessantes, como aumentar a estabilidade dos padrões de empacotamento e permitir o transporte de diferentes tipos de itens, com características específicas. Na literatura, em diversos estudos, o carregamento de contêineres multicompartimentados tem sido abordado juntamente com os problemas de roteamento de veículos, descrevendo inúmeras aplicações. A partir de uma revisão de textos que tratam destes problemas, Ostermeier et al. (2021) apontam uma formulação geral e uma tipologia, considerando aspectos relacionados aos compartimentos e as entregas. A partir da revisão de literatura, buscamos identificar os Problemas de Carregamento de Veículos Multicompartimentados (MVCP) em diferentes áreas de aplicação.

Como foco deste estudo, consideramos um problema de carregamento de contêineres multicompartimentados, em que as caixas são embaladas em caminhões compartimentados. Este problema foi apresentado inicialmente por Ranck Jr. (2014), que apresenta sua modelagem matemática e uma primeira proposta de resolução. Este problema consiste em realizar o carregamento de um conjunto de produtos de uma empresa da indústria de bebidas, distribuídos a um determinado grupo de clientes. A lista de entregas é pré-definida pela empresa, que apresenta as demandas de cada tipo de item e a sequência de paradas. O processo de entrega é realizado utilizando um dos tipos de veículo disponível na frota da firma, diferenciados pela quantidade de compartimentos em que a carroceria é dividida. Esta estrutura é de grande utilidade para os casos em que se busca ampliar a estabilidade ou garantir a divisão da carga no transporte. O percurso que o caminhão deve percorrer é previamente determinado e todos os itens requisitados pelos clientes necessitam estar acomodados no contêiner.

Ao alocar itens em objetos, uma série de restrições são observadas e precisam ser consideradas no cumprimento da atividade. De acordo com Bortfeldt e Wäscher (2013), as restrições podem ser classificadas como fortes, que precisam ser satisfeitas, e fracas, que ficam em segundo plano e podem ser violadas. No caso abordado, foram consideradas como forte as seguintes restrições: orientação dos itens, estabilidade de carga, resistência; e o balanceamento de peso no contêiner como fraca. Deste modo, o custo do desvio de balanceamento de peso deve ser minimizado, juntamente com o custo de remanejamento de caixas. O remanejamento considera necessário movimentar uma caixa quando ela estiver acomodada sobre, ou à frente, de itens de destinos anteriores a serem descarregados antes dela.

Para esta aplicação, Ranck Jr. (2014) apresenta uma heurística baseada na resolução de dois modelos de Programação Linear Inteira Mista. A resolução destes modelos pode custar um alto tempo computacional. A proposta deste trabalho consiste em implementar uma heurística construtiva, capaz de realizar o carregamento de uma lista de itens demandados. O objetivo almejado com esta proposta é obter soluções factíveis e de boa qualidade, em relação aos custos considerados na função objetivo do problema abordado, e em menor tempo de execução computacional.

O empacotamento dos itens ocorre por distribuir as caixas em camadas horizontais, classificadas em: completas e incompletas. As camadas completas têm um padrão de carregamento pré-definido, em que é considerado apenas a quantidade máxima de itens que este tipo de estrutura comporta. Para resolver esta parte do problema, propomos uma etapa de pré-processamento, que considera a construção das camadas completas como um problema de empacotamento bidimensional. Para isso, foi utilizado o modelo proposto por Beasley (1985), em conjunto com a discretização apresentada por Herz (1972) e Christofides e Whitlock (1977).

Para o empacotamento das camadas completas é necessário adotar um procedimento que permita uma boa distribuição das camadas completas nos compartimentos. Para isso, cada compartimento é interpretado como uma mochila unidimensional, em que camadas completas de diversas alturas devem ser combinadas de modo a não ultrapassar a altura do

compartimento. Deste modo, para cada compartimento foi resolvido um problema da mochila considerando que o valor de utilidade das camadas é igual a altura das camadas (*subset sum problem*)

O processo de construção das camadas incompletas é encarado como um problema de empacotamento tridimensional e foi baseado na proposta apresentada por Moura e Oliveira (2005), em que as caixas são acomodadas em paredes verticais. Os itens que não fazem parte das camadas completas são classificados como residuais, ordenados pela altura, em ordem decrescente, e destinados a compor as estruturas incompletas. A confecção de uma camada incompleta para um compartimento ocorre pela distribuição das caixas em pilhas, seguindo a lista de itens residuais. Um tipo de item é selecionado e uma pilha é montada com o máximo de caixas, respeitando a altura residual do compartimento. Esta pilha é replicada na direção horizontal, alocando o máximo de pilhas com a quantidade disponível de itens do tipo selecionado e respeitando a largura do compartimento. Os espaços acima e ao lado destas pilhas são verificados, na tentativa de construir novas pilhas, e o processo de construir pilhas se repete até que não seja possível acomodar mais itens nestes espaços, concluindo assim a montagem de uma parede. Este procedimento se repete, até que o compartimento seja completamente preenchido. Caso existam ainda itens residuais, o processo de construção de uma camada incompleta se repete para um novo compartimento, até que não se tenham mais itens a serem carregados. Vale destacar que todos os itens devem ser empacotados para que a solução seja factível.

O texto está organizado de maneira que o Capítulo 2 apresenta uma revisão da tipologia para os problemas de corte e empacotamento e uma revisão das principais restrições práticas consideradas nos problemas de carregamento de veículos. Ademais, apresenta uma revisão da tipologia para os problemas de roteamento de veículos multicompartimentados, bem como, as principais áreas de aplicação para estes problemas. O Capítulo 3 apresenta a definição para o problema de carregamento de veículo multicompartimentado considerado e sua modelagem matemática, estabelecida inicialmente por Ranck Jr. (2014). O Capítulo 4 apresenta o método de resolução proposto, descrevendo a distribuição dos itens em camadas horizontais e os modelos matemáticos para o problema de empacotamento bidimensional e problema da mochila, presentes nesta etapa. No Capítulo 5 são descritos os experimentos computacionais realizados, gerados a partir de um exemplo prático e no Capítulo 6, são apresentadas as considerações finais, oferecendo uma síntese dos principais pontos discutidos ao longo do trabalho.

2 PROBLEMAS DE CORTE E EMPACOTAMENTO: CLASSIFICAÇÃO E REVISÃO BIBLIOGRÁFICA

Neste capítulo, na Seção 2.1 é apresentada uma breve descrição dos critérios considerados na elaboração da tipologia para os PCE. Na Seção 2.2 é realizada uma revisão da classificação estabelecida para os PCE em relação aos casos uni, bi e tridimensional. Ademais, apresenta também uma revisão das principais restrições práticas consideradas nos problemas de carregamento de contêineres.

Na sequência do capítulo, na Seção 2.3 é apresentada a revisão de estudos acerca dos problemas de carregamento de veículos multicompartimentados (MVCP - *multi-compartment vehicle loading problems*) nos casos uni, bi e tridimensional. Além disso, é apresentada uma revisão da tipologia para os problemas de roteamento de veículos (MCVRP - *Multi-Compartment Vehicle Routing Problem*). Bem como uma revisão das principais áreas de aplicação destes problemas, com destaque para área de distribuição de alimentos.

2.1 TIPOS DE PROBLEMAS

Problemas de corte e empacotamento são amplamente estudados devido as suas aplicações em diversos ramos industriais. Em razão da grande variedade de problemas de corte e empacotamento existentes, Dyckhoff e Finke (1992) propuseram uma primeira tipologia que está baseada nos critérios de dimensionalidade, tipo de atribuição, variedade de objetos e variedade de itens.

Conforme a necessidade, Wäscher, Haußner e Schumann (2007) elaboraram uma tipologia a partir da apresentada por Dyckhoff e Finke, buscando obter uma categorização mais condizente com a realidade. Alguns critérios e a notação foram alterados e, ademais, a forma dos itens é proposta como um novo critério. Temos então os seguintes critérios:

1. Dimensionalidade: sendo a quantidade de dimensões relevantes para o problema, podendo ter uma, duas, três ou mais dimensões.
2. Tipo de Atribuição: dado pela maximização da saída (por exemplo, maximizar o valor dos itens empacotados nos objetos) e minimização de entrada (por exemplo, minimizar a quantidade de objetos utilizados). No primeiro caso, a quantidade de objetos não é suficiente para empacotar ou cortar todos os itens, enquanto que no segundo caso a quantidade de objetos é suficiente.
3. Variedade de itens: divididos em idênticos, pouco heterogêneos, muito heterogêneos.
4. Variedade de objetos: um objeto ou vários objetos. No caso de um único objeto, ele que pode ter todas as dimensões fixas ou uma ou mais dimensões variáveis. Para o caso de

vários objetos, eles têm todas as dimensões fixas e podem ser classificados de acordo com a variedade, assim como descrito para os itens.

5. Forma de itens: podendo ser itens regulares (por exemplo, retângulos, círculos, caixas, esferas) ou irregulares (qualquer formato).

Utilizando os critérios (2) e (3), Wäscher, Haußner e Schumann (2007) apresentam uma classificação mais elaborada e lista inicialmente tipos básicos de problemas sobre corte e empacotamento. A partir deste processo, buscando por problemas mais homogêneos, é introduzido o critério (4) e assim, propostos os tipos intermediários, visando a maximização da saída (produção de itens) e a minimização da entrada (utilização de objetos). Por fim, acrescentando os critérios (1) e (5), são definidos os tipos de problemas refinados.

2.2 PROBLEMAS DE CORTE E EMPACOTAMENTO

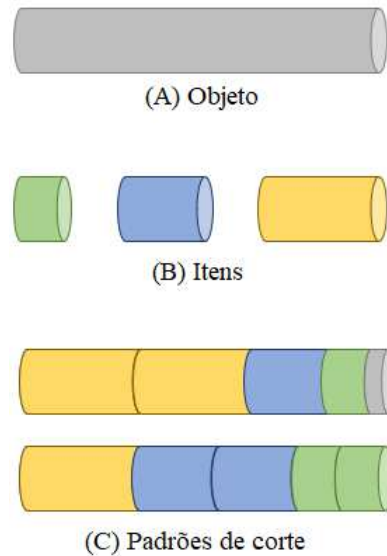
Nos problemas de corte e empacotamento, os itens podem ser acomodados de diversas maneiras em um objeto, de modo que as restrições básicas de não sobreposição, isto é, dois itens não podem ocupar o mesmo espaço, e de que os itens devem estar inteiramente contidos no objeto devem ser satisfeitas. A maneira que os itens são alocados em um objeto define um padrão de corte. Nestes problemas, um critério que apresenta grande significância é a dimensionalidade, necessário para descrever a geometria dos padrões de corte e empacotamento.

O problema unidimensional é denominado assim, pois, apenas uma dimensão do objeto é relevante para o processo de corte, em que o propósito é obter itens de tamanhos específicos a partir de um objeto, como observado na Figura 2.1. Um exemplo deste problema pode ser encontrado na indústria de papel, em que bobinas grandes (objetos) são cortadas em bobinas de menor comprimento e mesmo diâmetro (itens). Estudos acerca de problemas de corte e empacotamento para o caso unidimensional são encontrados em Kantorovich (1960), Gilmore e Gomory (1961), Stadtler (1990), Poldi (2003) e Kallrath et al. (2014).

Em problemas bidimensionais, duas dimensões relevantes são consideradas para o processo de corte e podem apresentar padrões de cortes com diversas configurações. Esses podem ser classificados como regulares (itens e objetos retangulares ou circunferenciais), ou irregulares. No caso dos problemas de corte bidimensional em que os itens são retangulares, os padrões de corte são divididos em guilhotinados e não-guilhotinados. Nos padrões guilhotinados, cada corte é realizado de um lado ao outro do objeto, gerando dois novos objetos, como ilustrado na Figura 2.2 (C). Enquanto que, os padrões não-guilhotinados não podem ser obtidos através de cortes guilhotinados, conforme observado na Figura 2.2 (D).

Este tipo de problema pode ser observado em indústrias metalúrgicas, moveleiras, têxtil e de couro, entre outras. Em muitas aplicações, os itens serão utilizados na composição de algum produto, não sendo apenas peça final da produção. Algumas aplicações

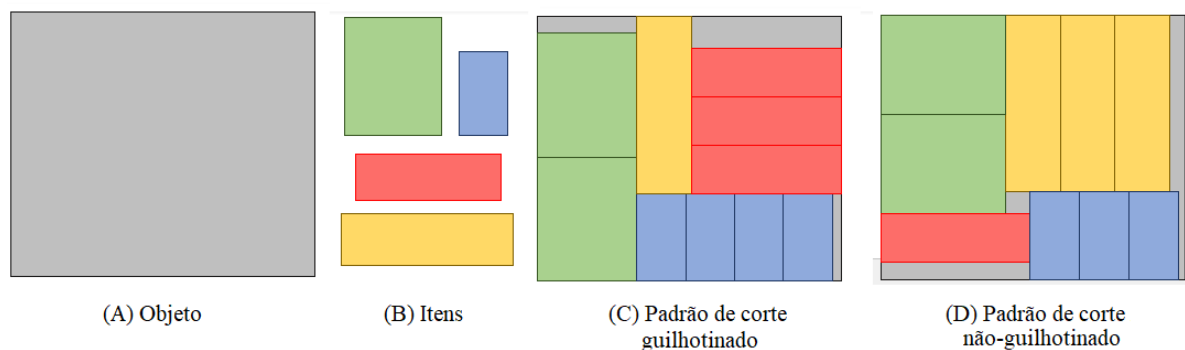
Figura 2.1: Problema de corte unidimensional.



Fonte: Adaptado de Bezerra (2018).

e modelos matemáticos para o caso bidimensional dos problemas de corte e empacotamento podem ser observados em Gilmore e Gomory (1965), Lodi, Martello e Monaci (2002), Ghidini (2009) e Bezerra et al. (2020).

Figura 2.2: Problema de corte bidimensional.



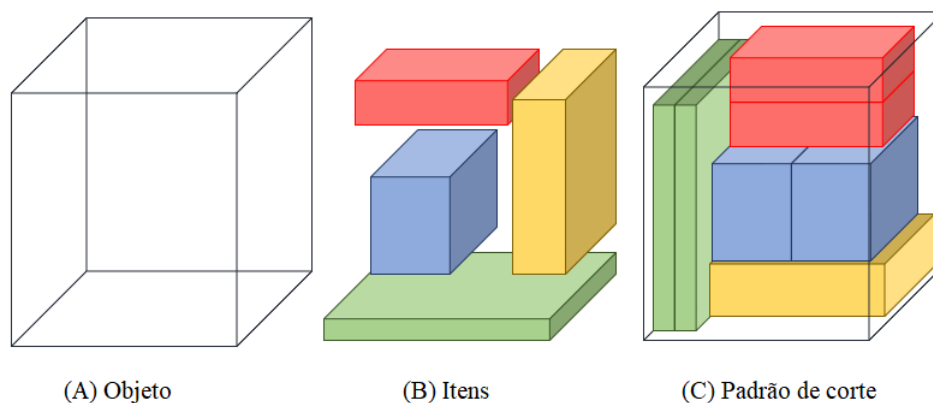
Fonte: Adaptado de Bezerra (2018).

Para o caso tridimensional, três dimensões são relevantes no processo de corte e empacotamento, como ilustrado na Figura 2.3. Algumas semelhanças com o caso bidimensional podem ser observadas, como a grande variedade de padrões de corte e a distinção em guilhotinados e não-guilhotinados. Algumas discussões acerca de problemas para o caso tridimensional são encontradas em Zhu e Lim (2012), Silva, Toffolo e Wauters (2019), Silva et al. (2020) e Gajda et al. (2022).

Para o caso guilhotinado, observado em problemas de corte, uma aplicação pode ser encontrada em fábricas de travesseiros e colchões, que utilizam espumas como matéria-prima para o preenchimento dos produtos. O caso não-guilhotinado, se observado em problemas

de carregamento de contêineres, um exemplo pode ser encontrado no setor de logística em atividades de armazenamento, carregamento e transporte de itens em unidades maiores.

Figura 2.3: Problema de corte tridimensional.



Fonte: Adaptado de Bezerra (2018).

Para esses problemas, além das propriedades geométricas básicas consideradas, uma série de outras restrições práticas são necessárias a fim de descrever melhor um problema real. Bortfeldt e Wäscher (2013) apresentaram uma revisão dos estudos presentes na literatura, buscando identificar as principais restrições consideradas em problemas de carregamento de contêineres, e que são descritas a seguir.

- **Orientação:** define a disposição que o item pode ser carregado, como o lado que deve ficar para cima ou para quais lados a caixa pode ser rotacionada. Em alguns casos a orientação é pré-definida.
- **Resistência ao empilhamento:** considera a resistência dos itens e determina a quantidade de caixas que podem ser alocadas uma sobre as outras para evitar que as mercadorias se danifiquem.
- **Manuseio:** garante que alguns itens ocupem determinadas posições no contêiner devido às suas dimensões, ao seu peso ou ao equipamento utilizado no carregamento.
- **Estabilidade:** verifica a estabilidade vertical observando o apoio da base dos itens, e a horizontal considerando a área de apoio das laterais dos itens. Para garantir a estabilidade durante todo o percurso, podem ser utilizados suportes ou materiais de enchimento.
- **Alocação da carga:** garante que uma seleção de itens que precisam ser alocadas juntos sejam carregados no mesmo compartimento (restrições de conectividade), ou para que itens que não podem ser acomodados próximos sejam carregados separados (restrições de separação).

- **Múltiplos destinos:** considera a sequência de entrega dos itens para diversos destinos diferentes para evitar operações de carga e descarga desnecessárias.
- **Envio completo:** para os problemas em que o número de contêineres é insuficiente, determinados itens não são carregados, nesta situação, se um item é carregado, todos os itens do mesmo conjunto também devem ser carregados (objetos que apresentam uma relação ou dependência).
- **Prioridades:** seleciona os itens a serem carregados, quando a quantidade de contêineres é insuficiente, estabelecendo prioridades de carregamento como prazos de entrega ou validade dos produtos.
- **Complexidade do padrão:** verifica as limitações para acomodar os padrões de carregamento, que podem não ser aceitáveis para o carregamento manual, seja pela dificuldade ou tempo de operação. Com a utilização de tecnologias mecânicas, padrões complexos podem gerar a necessidade de mão de obra adicional para auxiliar no carregamento, elevando os custos da operação.
- **Limite de peso:** garante que o peso da carga não extrapole o limite de peso suportado pelo contêiner.
- **Balanceamento de peso:** distribui igualmente o peso da carga pelo contêiner para diminuir a possibilidade de deslocamento da carga durante o transporte. Além disso, estas restrições podem garantir que o centro de gravidade da carga esteja localizado próximo do centro geométrico do plano que define a base do contêiner.

No processo de carregamento de contêineres, as restrições podem ser consideradas como rígidas (fortes) e que devem ser satisfeitas, caso contrário, o padrão de carregamento será infactível, ou como suaves (fracas) em que as restrições devem ser mantidas, porém violações limitadas são toleráveis.

Os problemas de carregamento de contêineres apresentam grande relevância prática entre os quais pode ser observado o problema de carregamento de contêineres multicompartimentados (MCP - *Multi-compartment Packing Problem*), nos quais o espaço de carga dos veículos pode ser dividido em compartimentos, auxiliando assim na distribuição de diferentes tipos de itens (OSTERMEIER et al., 2018).

2.3 PROBLEMAS DE CARREGAMENTO DE VEÍCULOS MULTICOMPARTIMENTADOS

Problemas de corte e empacotamento compartimentados podem surgir nos casos unidimensional, bidimensional e tridimensional. No caso unidimensional, os itens são separados em classes de acordo com alguma característica e um compartimento deve ser composto apenas por itens da mesma classe. Este problema surge no corte de bobinas de aço,

em que uma bobina grande é cortada em bobinas intermediárias, cujo comprimento deve estar dentro de um intervalo. Cada bobina intermediária passa por um processo de laminação para a redução de espessura, que deve coincidir com a espessura de uma determinada classe de itens. Por fim, cada bobina intermediária é cortada em bobinas menores (itens finais) de uma determinada classe. Deste modo, o corte dos objetos é realizado em duas fases. Estudos para o problema de corte compartimentado unidimensional podem ser encontrados em Leão et al. (2011), Quiroga-Orozco, Carvalho e Hoto (2019) e Borges et al. (2020).

O problema compartimentado bidimensional é uma extensão do caso unidimensional. Como sugerido em Leão (2013), possíveis aplicações para este problema podem ser encontradas em problemas cuja placa original é cortada em placas menores que, por sua vez, recebem algum tipo de tratamento (banhos químicos ou pinturas) e, então, são recortadas em itens retangulares menores.

Entre os problemas de empacotamento tridimensionais encontram-se os problemas de carregamento de veículos multicompartimentados, em que o veículo é dividido em compartimentos. Estes problemas consistem em realizar o carregamento de uma variedade de itens em um conjunto de veículos multicompartimentados, de maneira a minimizar determinados objetivos.

Veículos multicompartimentados podem ser úteis em diversas situações e problemas de carregamento que utilizam este tipo de contêiner têm sido observados principalmente em estudos acerca do roteamento de veículos (MCVRP - *Multi-Compartment Vehicle Routing Problem*). Estes problemas buscam determinar a rota a ser percorrida por uma frota de veículos na qual as entregas devem ser realizadas, respeitando a capacidade dos contêineres durante o carregamento dos pedidos, garantindo o atendimento total das demandas e buscando a minimização dos custos da viagem.

Diversos MCVRPs são discutidos na literatura e diferentes estudos podem ser desenvolvidos a partir de inúmeros tipos de aplicações. Ostermeier et al. (2021) apresentam uma revisão de publicações acerca de MCVRPs, destacando uma formulação geral e propondo uma tipologia para estes problemas. Para montar uma tipologia, diferentes características observadas nestes problemas foram avaliadas, sendo considerados os aspectos relacionados aos compartimentos (flexibilidade, atribuição dos itens e compartilhamento) e as entregas (número de visitas e modo de atendimento da demanda), como podem ser observadas na Figura 2.4 e descritas a seguir.

1. **Flexibilidade:** o tamanho dos compartimentos pode ser fixos (FixS - *Fixed Size*) ou flexíveis (FlexS - *Flexible Size*), o que pode determinar a quantidade de compartimentos no contêiner;
2. **Atribuição dos itens:** a distribuição dos itens nos compartimentos pode ser fixa (FixA - *Fixed Assignment*), em que cada compartimento é destinado a um tipo de item, ou flexível (FlexA - *Flexible Assignment*), em que qualquer item pode ser acomodado em qualquer

um dos compartimentos;

3. **Compartilhamento:** o carregamento dos itens pode ocorrer de maneira que cada compartimento seja destinado a acomodar um tipo de item, com a demanda de apenas um cliente (UC - *Unshared Compartments*), ou um compartimento pode acomodar um tipo de item agrupando a demanda de vários clientes (SC - *Shared Compartments*);
4. **Número de visitas:** a entrega dos pedidos de um cliente deve ser com uma visita única (SV - *Single Visits*), em que os diferentes tipos de item demandados são entregues de uma vez, utilizando apenas um veículo, ou ainda, em múltiplas visitas (MV - *Multiple Visits*) em que os diferentes tipos de produtos demandados são entregues por veículos distintos, até o atendimento total da demanda solicitada;
5. **Modo de atendimento da demanda:** a entrega de um tipo de item demandado por um cliente não deve ser dividida, ou seja, deve ser entregue de uma vez e utilizando apenas um veículo (UD - *Unsplit Demand*), ou então, pode ser dividida (SD - *Split Demand*), em que não somente diferentes tipos de itens podem ser entregues por diferentes veículos, mas também, a demanda de um tipo de item pode ser dividida entre diferentes veículos.

Figura 2.4: Tipologia para MCVRP.



Fonte: Adaptado de (OSTERMEIER et al., 2021).

Contêineres com múltiplos compartimentos não perdem a capacidade total de carga pela utilização de divisórias, o espaço disponível para carregamento é dividido entre os compartimentos que compõem o objeto. As dimensões dos compartimentos e a quantidade dos compartimentos podem ser fixas ou flexíveis, em que neste último caso, as divisórias são ajustáveis de acordo com a necessidade no empacotamento dos itens.

A diversidade dos itens que compõem as demandas destacam a versatilidade que veículos multicompartimentados podem proporcionar, visto que os itens podem ser alocados em qualquer um dos compartimentos. Além disso, a divisão do veículo em compartimentos

garante a separação de alguns produtos que não podem ser acomodados juntos (RANCK Jr., 2014).

Em sua revisão, Ostermeier et al. (2021) destacam a diversidade das aplicações presentes na literatura, agrupando os problemas em grandes áreas que tratam da distribuição de combustíveis, coleta de resíduos, transporte agrícola, distribuição de alimentos, transporte marítimo e outras áreas de aplicação. A seguir é apresentada uma descrição de alguns problemas relacionados às seguintes áreas citadas.

Distribuição de combustíveis - Problemas relacionados a distribuição de combustíveis são apontados como os primeiros a considerar a utilização de veículos com múltiplos compartimentos (MCV - *Multi-compartment Vehicle*), em que um ou diversos tipos de combustíveis precisam ser transportados das refinarias ou distribuidoras até seus clientes. Nestes problemas, os autores observam a utilização única de veículos com compartimentos fixos (FixS), com os produtos podendo ser atribuídos a qualquer um dos compartimentos. Além disso, os problemas adotam com frequência a distribuição flexível dos tipos de combustíveis nos compartimentos (FlexA), bem como o não compartilhamento dos compartimentos com demandas de vários clientes (UC) e as entregas sendo realizadas em visitas únicas (SV). Algumas discussões envolvendo este setor podem ser encontradas em Brown et al. (1987) e Cornillier et al. (2008).

Coleta de resíduos - Problemas encontrados no contexto da coleta de resíduos consideram a utilização de MCVs com bastante frequência, em que diferentes tipos de produtos são apanhados em clientes, ou pontos de coleta, e que em muitos casos não podem ser misturados durante o transporte. Em relação a tipologia, os autores relatam emprego da atribuição fixa dos tipos de itens aos veículos (FixA), bem como o compartilhamento dos compartimentos com resíduos coletados em mais de um cliente (SC). Ademais, a utilização de compartimentos de tamanho fixo (FixS) e a coleta sendo realizada com múltiplas visitas (MV) são frequentemente observadas. Algumas aplicações para este setor podem ser encontradas em Muyldermans e Pang (2010) e Elbek e Wøhlk (2016).

Transporte na agropecuária - Problemas que abordam o transporte de produtos do setor agrícolas, como leite, azeite, animais ou ração animal, consideram com regularidade a utilização de veículos com compartimentos de quantidade e tamanho fixos (FixS). O carregamento dos produtos ocorre, em sua maioria, de maneira flexível (FlexA) e sem que as demandas sejam divididas (UD). Diversas vezes ocorre o compartilhamento do compartimento com diferentes demandas (SC), e as visitas ocorrem de maneira única, na maioria dos casos (SV). Algumas aplicações para este setor podem ser obtidas em Ruiz, Maroto e Alcaraz (2004) e Oppen e Løkketangen (2008).

Distribuição de alimentos - Problemas associados ao setor varejista mostram as contribuições da utilização de MCV na distribuição de mantimentos, possibilitando o transporte eficiente de produtos variados. Aplicações são vistas em supermercados ou lojas de conveniência, que comercializam itens congelados, alimentos frescos (como carne e laticínios) e itens diários em temperatura ambiente (como artigos domésticos, cosméticos e de higiene). Estão implícitos no processo de distribuição o compartilhamento dos compartimentos (SC) e a não divisão das demandas (UD). A modernidade dos veículos de transporte permitem a utilização de compartimentos flexíveis (FlexS), em muitos casos que é necessário criar zonas de temperaturas. Produtos possuem requisitos específicos e com isso, a distribuição nos compartimentos é sempre fixa (FixA). Por fim, são encontrados estudos que consideram as entregas tanto em visitas únicas (SV) como em múltiplas (MV). Algumas discussões acerca desta aplicação podem ser encontradas em Eglese, Mercer e Sohrabi (2005) e Martins et al. (2019).

Transporte Marítimo - Em problemas no âmbito marítimo, é recorrente a utilização de múltiplos compartimentos por grandes navios que transportam diferentes tipos de produtos. Observando os atributos relacionados a estes problemas, em grande parte dos estudos considera-se a utilização de compartimentos com tamanhos fixos (FixS). Além disso, também ocorre o carregamento compartilhado da demanda de diferentes clientes (SC) e a atribuição flexível dos itens aos compartimentos (FlexA). Finalmente, em grande parte são consideradas múltiplas visitas (MV), bem como demandas não divididas (UD) no abastecimento dos portos. Problemas de roteamento de navios (SRPs). Algumas aplicações podem ser observadas em Bausch, Brown e Ronen (1998) e Siswanto, Essam e Sarker (2011)

Além das áreas citadas, outras aplicações podem ser encontrados na literatura, em que são analisados contextos específicos como sistemas de partilha de bicicletas, distribuição de produtos químicos, transporte de passageiros ou até mesmo problemas sem uma aplicação explícita, destacam Ostermeier et al. (2021). Podemos observar que algumas aplicações podem envolver o problema de empacotamento tridimensional, principalmente problemas aplicados à distribuição de alimentos. A seguir revisamos alguns destes trabalhos.

Ostermeier et al. (2018) abordam um problema de carregamento de itens idênticos no setor de varejo, a partir da logística de distribuição de alimentos. A utilização de veículos refrigerados com múltiplos compartimentos e de tamanhos flexíveis torna-se relevante visto a necessidade de diferentes níveis de temperatura para o transporte dos produtos, por exemplo: itens congelados, frescos e em temperatura ambiente. O problema consiste em definir a rota de entrega dos produtos e os tamanhos dos compartimentos, o carregamentos dos itens para obter rotas viáveis e com o menor custo. Como proposta de resolução, os autores desenvolvem duas abordagens, uma a partir do método exato de programação linear inteira *Branch-and-cut*, a fim de obter um solução para o problema de carregamento, e uma heurística de busca em vizinhança de grande porte (LNS - *Large Neighborhood Search*), para melhoramento da solução inicial.

Ranck Jr. et al. (2019) apresentam um MCVRP observado no dia a dia de trabalho de uma empresa de refrigerantes que realiza entregas de diferentes tipos de produtos para seus clientes, utilizando vários tipos de veículos multicompartimentos de tamanhos fixos, diferenciados pela quantidade de compartimentos. O carregamento é realizado seguindo a demanda de uma lista de entregas pré-definida. Buscando encontrar um plano de carregamento viável, a fim de minimizar a movimentação de produtos ao longo da rota de entrega, é proposta uma abordagem híbrida a partir da construção de camadas horizontais, tomando o melhor empacotamento gerado pelas heurísticas *First Fit* e *Best Fit*, associado à resolução de modelos de programação linear inteira mista utilizando o método *Branch-and-Cut*.

Os artigos a seguir, mesmo não tendo como foco principal um problema de empacotamento, entram como parte da revisão, uma vez que existe um problema de empacotamento envolvido na situação analisada.

Derigs et al. (2011) retratam um problema de roteamento de veículos compartimentados, observando situações teóricas do setor de varejo e de combustíveis, transportando produtos da indústria até os locais de comércio. O problema consiste em atribuir os pedidos aos veículos e seus compartimentos, de maneira a montar uma rota de entrega, a fim de minimizar os custos de transporte. São considerados veículos homogêneos, de mesma capacidade, e compartimentos fixos e flexíveis. Para resolução, os autores destacam diversas abordagens conhecidas para resolução deste tipo de problema e buscam identificar boas combinações entre heurísticas e estratégias de melhoramento. Para esta investigação, os autores organizam os métodos nas seguintes etapas: construção, meta-heurísticas, adaptação, busca em grandes vizinhanças e busca local.

Pirkwieser, Raidl e Gottlieb (2011) apresentam um problema de roteamento de veículos compartimentados, considerando também o processo de empacotamento, acerca da distribuição de diferentes tipos de combustíveis. Busca-se minimizar os custos finais do roteamento de veículos e também utilizar de maneira eficiente o espaço de carga disponível no carregamento. Os autores adotam uma resolução a partir de um método heurístico baseado na busca em vizinhança variável (*VNS - Variable Neighborhood Search*).

Hübner e Ostermeier (2018) destacam que o carregamento de veículos refrigerados multicompartimentados necessitam de um tempo operacional maior, vista a necessidade de carregar itens armazenados em áreas distintas devido as zonas de temperatura necessárias. Com isso, os autores apresentam uma resolução, a fim de determinar um itinerário, minimizando os custos de entrega (carga, transporte e descarga, desenvolvendo uma heurística baseada no algoritmo *Savings*, para gerar de maneira rápida uma solução inicial factível. Posteriormente, é desenvolvida uma heurística LNS, utilizada no processo de melhoramento da solução inicial.

No mesmo setor, Chang (2022) aborda um MCVRP com veículos refrigerados e compartimentos flexíveis para distribuição de diversos tipos de produtos com diferentes requisitos de processamento, como temperatura e umidade. Os compartimentos são ajustados

de acordo com a necessidade dos produtos demandados e a tomada de decisões ao alocar itens deve considerar os custos de carregamento dos diferentes segmentos de produtos. A fim de determinar uma plano de carregamento dos itens e uma rota de entrega é proposto um modelo matemático, com a resolução desenvolvida utilizando um algoritmo adaptativo de vizinhança de grande porte (ALNS). A estabilidade e superioridade da solução é verificada a partir de um exemplo do problema.

A partir da revisão dos textos supracitados, foi possível estabelecer uma compreensão dos estudos desenvolvidos acerca de problemas de corte e empacotamento, com ênfase nos problemas de carregamento de veículos multicompartimentados. Além disso, permitiu estabelecer uma base teórica para definição do problema a ser estudado neste trabalho, bem como os métodos de resolução a serem considerados.

3 APLICAÇÃO ESTUDADA

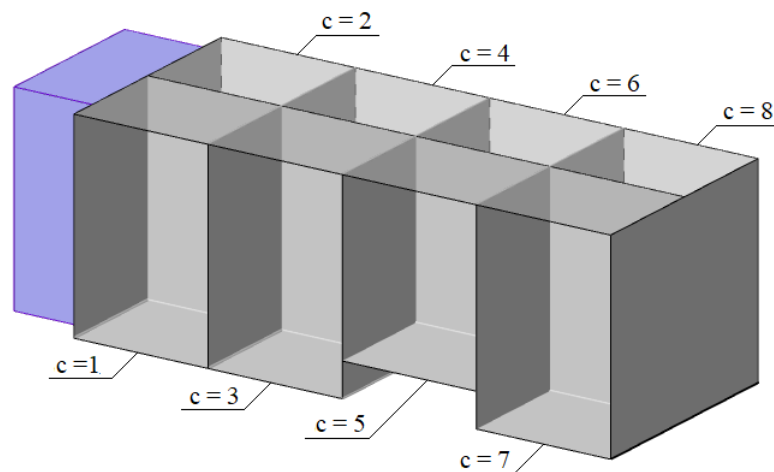
Neste capítulo será apresentada a aplicação considerada no desenvolvimento deste trabalho. Na Seção 3.1, será desenvolvida a definição do problema, um MVCP observado em um empresa de refrigerantes, descrito inicialmente por Ranck Jr. (2014). Na Seção 3.2 é descrita a modelagem matemática do problema considerado, descrevendo a função objetivo e as restrições práticas consideradas.

3.1 DEFINIÇÃO DO PROBLEMA

O problema de carregamento de veículo multicompartimentado pode ser encontrado em algumas empresas do setor de bebidas. Essas empresas são responsáveis pela produção, engarrafamento e distribuição de refrigerantes. A empresa dispõem de uma frota própria de veículos, diferenciados pela quantidade de compartimentos em que carrocerias dos caminhões são divididas.

Os contêineres são formados por compartimentos fixos, distribuídos em duas fileiras, acessados diretamente pelas laterais do veículo. Possuem dimensões de largura e comprimento iguais, mas alguns podem apresentar diferentes alturas, dependendo da posição do eixo do caminhão, conforme ilustrado na Figura 3.1.

Figura 3.1: Veículo multicompartimentado.

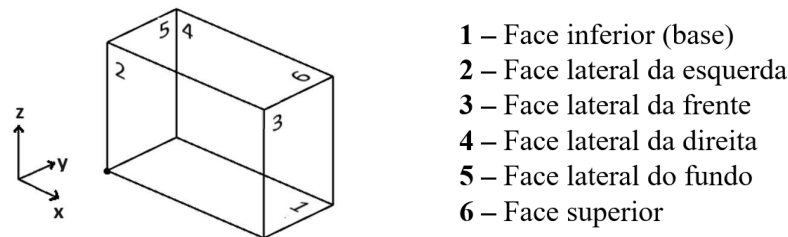


Fonte: Adaptado de Ranck Jr. (2014).

O transporte dos produtos é realizado seguindo uma rota de entrega pré-definida pelo sistema de roteamento de veículos da empresa, que determina uma lista de clientes a serem visitados sequencialmente, bem como suas respectivas demandas para cada tipo de item. A lista de clientes inicia com a visita ao destino 1, em seguida a visita ocorre no destino 2, e assim por diante, realizando o carregamento completo da demanda de cada cliente.

As caixas e os compartimentos, são hexaedros regulares, e de acordo com os dados apresentados por Ranck Jr. (2014), são ofertados diferentes tipos de produtos, cada um com características específicas como: dimensão, peso, volume relativo e resistência ao empilhamento. Na Figura 3.2 é possível observar uma representação de um item, com a identificação adotada para as faces deste.

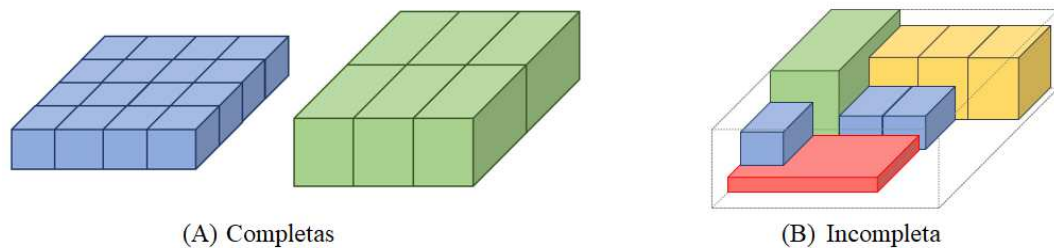
Figura 3.2: Item de referência com a nomenclatura de cada face.



Fonte: Adaptado de Ranck Jr. (2014).

O processo de carregamento dos compartimentos se desenvolve por acomodar os itens ortogonalmente em camadas horizontais. As camadas ditas completas, conforme pode ser observado na Figura 3.3 (A), são compostas por um mesmo tipo de item, acomodados seguindo uma mesma orientação, que possibilite alocar o máximo de itens na camada, sem empilhá-los. As camadas incompletas, como ilustrado na Figura 3.3 (B), são formadas por diferentes tipos de produtos, seguindo a melhor orientação para cada item e permitindo o empilhamento de caixas, desde que exista suporte e resistência de carga necessários.

Figura 3.3: Organização dos itens em camadas.

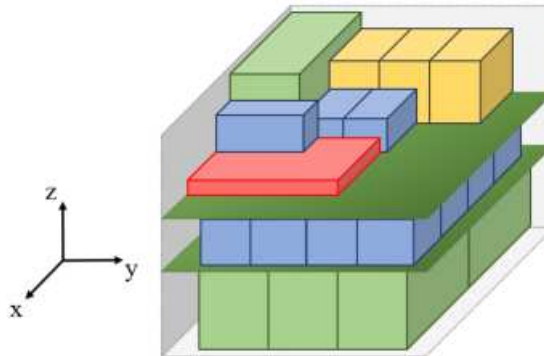


Fonte: Adaptado de Ranck Jr. (2014).

No carregamento, as camadas completas são acomodadas em posições mais abaixo nos compartimentos, de maneira que as camadas incompletas ocupem sempre a parte mais a cima no compartimento, sendo sempre a última a ser empacotada. Estas estruturas são separadas por uma divisória fina com as mesmas dimensões do compartimento, como ilustrado na Figura 3.4 (RANCK Jr., 2014).

O empacotamento dos itens no veículo é realizado em um depósito, ou célula logística, em que deve-se garantir o atendimento total dos produtos demandados na lista de entregas, considerando as seguintes restrições práticas:

Figura 3.4: Camadas acomodadas em um compartimento.



Fonte: Adaptado de Ranck Jr. (2014).

- a) **Orientação:** as caixas podem ser rotacionadas na base em 90° , porém a orientação vertical é fixa;
- b) **Estabilidade:** as caixas devem ser estabilizadas ao longo de toda a rota, por outros itens e/ou pelas paredes do compartimento;
- c) **Resistência ao empilhamento:** as caixas podem ser empilhadas, isto é, colocadas uma acima das outras, desde que os itens abaixo tenham resistência suficiente para sustentar a pressão aplicada pelos itens acima;
- d) **Balanceamento de peso:** busca-se a distribuição do peso no contêiner, nos eixos x e y , ao longo de toda rota de entrega.

A restrição em d é considerada como fraca e deve ser satisfeita conforme a possibilidade, permitindo que um carregamento não tão balanceado seja impedido de ser utilizado, e as demais restrições são tidas como fortes e devem ser obrigatoriamente satisfeitas.

A fim de completar a apresentação do problema e com base nas descrições apresentadas por Ranck Jr. (2014), algumas observações precisam ser consideradas, admitindo-se que:

- a) o remanejamento de itens em uma parada consiste em retirá-los do veículo, acomodando-os temporariamente em uma área fora do contêiner e, em seguida, após a entrega de todos os itens desta parada, carregá-los novamente no veículo;
- b) a posição dos itens acomodados no contêiner não pode ser modificada ao longo da rota de entrega nos eixos x e y , somente no eixo z , quando uma camada alocada abaixo destes itens é totalmente descarregada;
- c) qualquer item deve ser sempre alcançado, mesmo que seja necessário o remanejamento de outros produtos;

- d) todos os itens devem estar totalmente apoiados com suas faces inferiores em outros itens, camadas ou contêiner;
- e) o peso sobre uma camada é distribuído homogeneamente entre os itens desta camada;
- f) a resistência ao empilhamento é distribuída homogeneamente por sua face superior;
- g) a densidade de cada item é uniforme;
- h) o contêiner, os itens e as divisórias entre camadas, são consideradas perfeitamente rígidas;
- i) as divisórias entre os compartimentos e entre as camadas possuem espessuras desprezíveis.

Neste estudo, o objetivo é determinar um plano de carregamento para os produtos demandados na lista de clientes, a fim de minimizar o manuseio de itens nos compartimentos durante todo percurso de entrega, isto é, o remanejamento. Ademais, a boa distribuição de peso no contêiner é um objetivo secundário a ser alcançado, visto que a restrição de balanceamento é do tipo fraca.

Conforme as revisões realizadas, de acordo com a tipologia de Wäscher, Haußner e Schumann (2007), o problema poderia ser classificado como um problema de mochila, porém uma solução para o problema só é considerada factível se todos os itens demandados forem empacotados, o que pode ser difícil de alcançar. Além disso, a função objetivo minimiza o remanejamento de itens e os desvios de balanceamento, o que não se enquadra na maximização de saída ou minimização de entrada. Em relação a compartimentação do contêiner, segundo a tipologia de Ostermeier et al. (2021), este problema trata da distribuição de alimentos, adotando compartimentos de tamanhos fixos (FixS), atribuição flexível dos itens (FlexA) e compartilhamento do contêiner com as demandas de vários clientes (SC). Em relação ao processo de entregas, são realizadas visitas únicas (SV) e sem a divisão das demandas (UD).

3.2 MODELAGEM MATEMÁTICA

A seguir descrevemos um modelo de programação linear inteira mista proposto para o problema de carregamento de veículo multicompartimentado. Este modelo foi apresentado em Ranck Jr. (2014) e está baseado nas formulações de Junqueira (2009).

3.2.1 Posições para Empacotamento

Os compartimentos que compõem o veículo têm suas posições definidas a partir de um sistema de coordenadas determinado para o contêiner, conforme apresentado na Figura 3.5. Os pontos de referência para os compartimentos são estabelecidos no canto inferior posterior esquerdo, estabelecido na posição $(p, q, 0)$ do contêiner. Os compartimentos são numerados sequencialmente, iniciando pelo compartimento dianteiro esquerdo, de maneira que

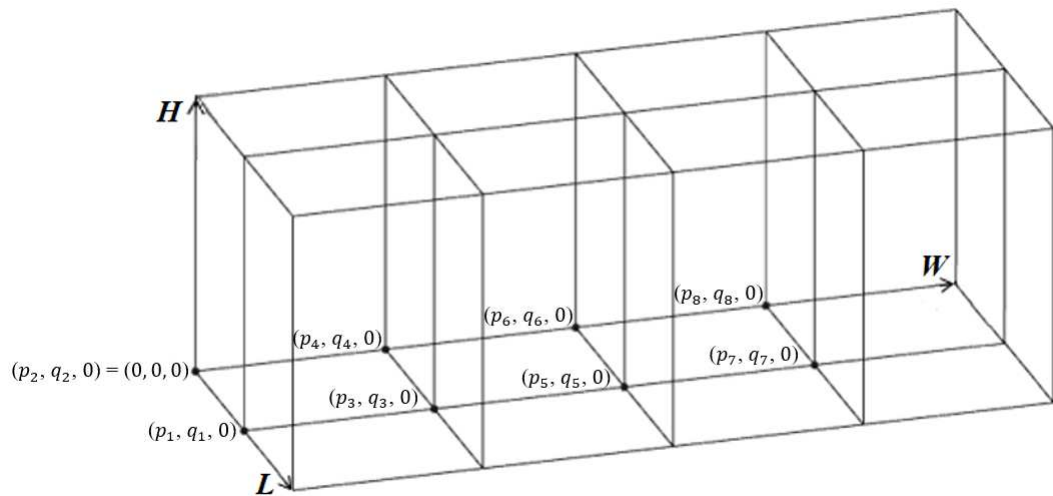
os compartimentos de números ímpares ficam todos do lado esquerdo do contêiner, e aqueles localizados do lado direito possuem números pares.

Os parâmetros considerados para o contêiner e os compartimentos são dados por:

- L, W, H : comprimento (eixo x), largura (eixo y) e altura (eixo z), respectivamente, do contêiner;
- C : conjunto dos compartimentos que compõem o contêiner;
- p_c, q_c : posições nos eixos x e y , respectivamente, que o compartimento $c \in C$ está posicionado no contêiner;
- L^{comp}, W^{comp} : comprimento (eixo x) e largura (eixo y), respectivamente, de todos os compartimentos;
- H_c^{comp} : altura (eixo z) do compartimento $c \in C$.
- M : conjunto dos tipos de itens demandados;
- M' : conjunto dos itens residuais;
- \bar{M} : conjunto dos tipos de itens e suas respectivas orientações de empacotamento ($\bar{M} \supset M$);
- b_i : Quantidade disponível (ou a demanda) do item de tipo $i \in M$;
- i, j : índice dos tipos de itens ou camadas;
- p, p' : índice para posição em relação ao eixo x , do item $i \in M$, em que $p, p' \in X_i^0$;
- q, q' : índice para posição em relação ao eixo y , do item $i \in M$, em que $q, q' \in Y_i^0$;
- r, r' : índice para posição em relação ao eixo z , do item $i \in M$, em que $r, r' \in Z_{ic}^0$;
- $f_{id}^{i,j}$: proposição lógica tida como verdadeira se o tipo de item $j \in \bar{M}$ e o tipo de item $i \in M$ são iguais, podendo os itens estarem em orientações distintas;
- ξ_j : variável auxiliar, em que $\xi_j \in \mathbb{N}, j \in \bar{M}$.

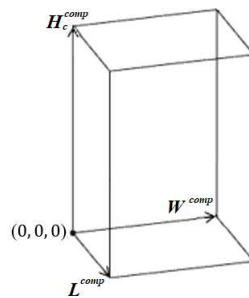
Assim como o contêiner, os compartimentos possuem um sistema de coordenadas próprio, com suas posições de referências definidas novamente no canto inferior posterior esquerdo, conforme pode ser observado na Figura 3.6. Na etapa de carregamento dos produtos, os itens são acomodados seguindo o sistema definido para o compartimento que está sendo carregado.

Figura 3.5: Sistema de coordenadas para um contêiner com oito compartimentos.



Fonte: Adaptado de Ranck Jr. (2014).

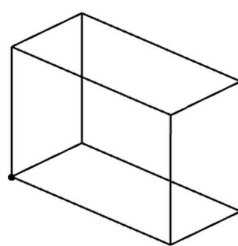
Figura 3.6: Sistema de coordenadas para um compartimento.



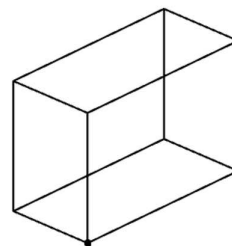
Fonte: Adaptado de Ranck Jr. (2014).

Como o carregamento das caixas ocorre de maneira ortogonal a base do contêiner, os itens têm sua orientação vertical fixa em relação ao plano e a horizontal variável, podendo assumir duas posições, ilustradas na Figura 3.7, em que as posições de empacotamento possíveis para os itens $i \in M$ são obtidas por rotacionar o item em 90° na horizontal.

Figura 3.7: Exemplo de possíveis orientações para o empacotamento de um item.



(A) Orientação 1



(B) Orientação 2

Fonte: Adaptado de Ranck Jr. (2014).

Adotando uma orientação, um item pode ocupar uma posição p , q e r no

compartimento $c \in C$, representada nos domínios X^0 , Y^0 e Z_c^0 , respectivamente, para os eixos x , y e z , sendo:

$$\begin{aligned} X^0 &= \{p : 0 \leq p \leq L^{comp} - \text{Min}_{i \in \bar{M}}(l_i), p \in \mathbb{N}\} \\ Y^0 &= \{q : 0 \leq q \leq W^{comp} - \text{Min}_{i \in \bar{M}}(w_i), q \in \mathbb{N}\} \\ Z_c^0 &= \{r : 0 \leq r \leq H_c^{comp} - \text{Min}_{i \in \bar{M}}(h_i), r \in \mathbb{N}\}, c \in C. \end{aligned} \quad (3.1)$$

Ao considerar um certo tipo de item $i \in \bar{M}$, os domínios definidos em (3.1) podem ser escritos como:

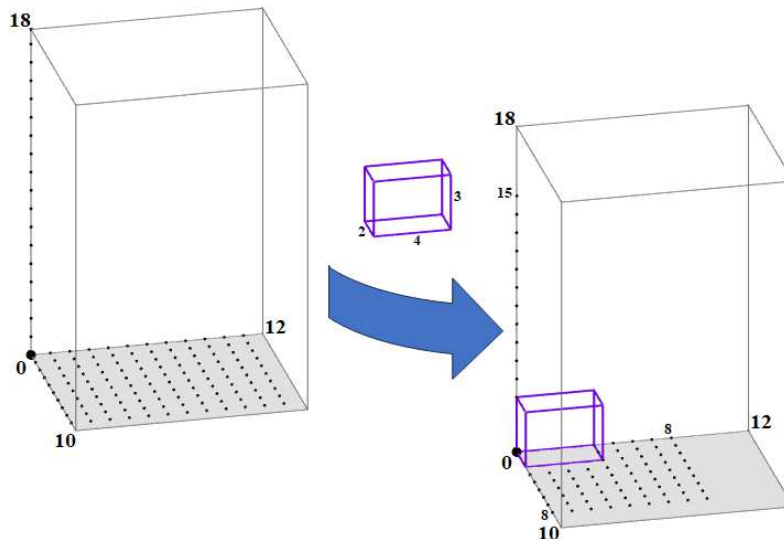
$$\begin{aligned} X_i^0 &= \{p : 0 \leq p \leq L^{comp} - l_i, p \in \mathbb{N}\}, i \in \bar{M} \\ Y_i^0 &= \{q : 0 \leq q \leq W^{comp} - w_i, q \in \mathbb{N}\}, i \in \bar{M} \\ Z_{ic}^0 &= \{r : 0 \leq r \leq H_c^{comp} - h_i, r \in \mathbb{N}\}, i \in \bar{M}, c \in C. \end{aligned} \quad (3.2)$$

Considerando, como exemplo, um compartimento $c \in C$, de dimensões $(L, W, H) = (10, 12, 18)$ e um item $i \in \bar{M}$ com dimensões $(l_i, w_i, h_i) = (2, 4, 3)$. Para este item i , os conjuntos X_i^0 , Y_i^0 e Z_{ic}^0 são dados por:

$$\begin{aligned} X_i^0 &= \{p : 0 \leq p \leq 8, p \in \mathbb{N}\} \\ Y_i^0 &= \{q : 0 \leq q \leq 8, q \in \mathbb{N}\}, \\ Z_{ic}^0 &= \{r : 0 \leq r \leq 15, r \in \mathbb{N}\}, c \in C. \end{aligned} \quad (3.3)$$

Na Figura 3.8 pode-se observar uma ilustração, para fins explicativos, dos domínios definidos em (3.3), com as possíveis posições (p, q, r) que um item $i \in \bar{M}$ pode ser acomodado no compartimento $c \in C$.

Figura 3.8: Posições para carregamento dos itens nos compartimentos.



Fonte: o Autor

Conforme constataram Herz (1972) e Christofides e Whitlock (1977), em um determinado padrão de empacotamento, as caixas podem ser movimentadas para baixo, para frente e/ou para esquerda, até que suas faces inferior, frontal e lateral esquerda entrem em contato com outras caixas, ou com o contêiner. Com isso, é possível reduzir os conjuntos definidos em (3.1) para:

$$\begin{aligned}
 X^0 &= \left\{ p : p = \sum_{j \in \bar{M}} \xi_j l_j, 0 \leq p \leq L^{comp} - \min_{i \in \bar{M}}(l_i), 0 \leq \sum_{\substack{j \in \bar{M}, \\ \text{se } J_{id}^{i,j}}} \xi_j \leq b_i, i \in M \right\} \\
 Y^0 &= \left\{ q : q = \sum_{j \in \bar{M}} \xi_j w_j, 0 \leq q \leq W^{comp} - \min_{i \in \bar{M}}(w_i), 0 \leq \sum_{\substack{j \in \bar{M}, \\ \text{se } J_{id}^{i,j}}} \xi_j \leq b_i, i \in M \right\} \\
 Z_c^0 &= \left\{ r : r = \sum_{j \in \bar{M}} \xi_j h_j, 0 \leq r \leq H^{comp} - \min_{i \in \bar{M}}(h_i), 0 \leq \sum_{\substack{j \in \bar{M}, \\ \text{se } J_{id}^{i,j}}} \xi_j \leq b_i, i \in M \right\}.
 \end{aligned} \tag{3.4}$$

O conjunto Z_c^0 é definido para cada compartimento $c \in C$. Os padrões que são gerados ao considerar os conjuntos reduzidos (3.4) são chamados de padrões normais. Contudo, usar esses padrões pode levar a uma perda da generalidade da solução do problema, quando considerado junto as restrições de estabilidade definidas mais a frente.

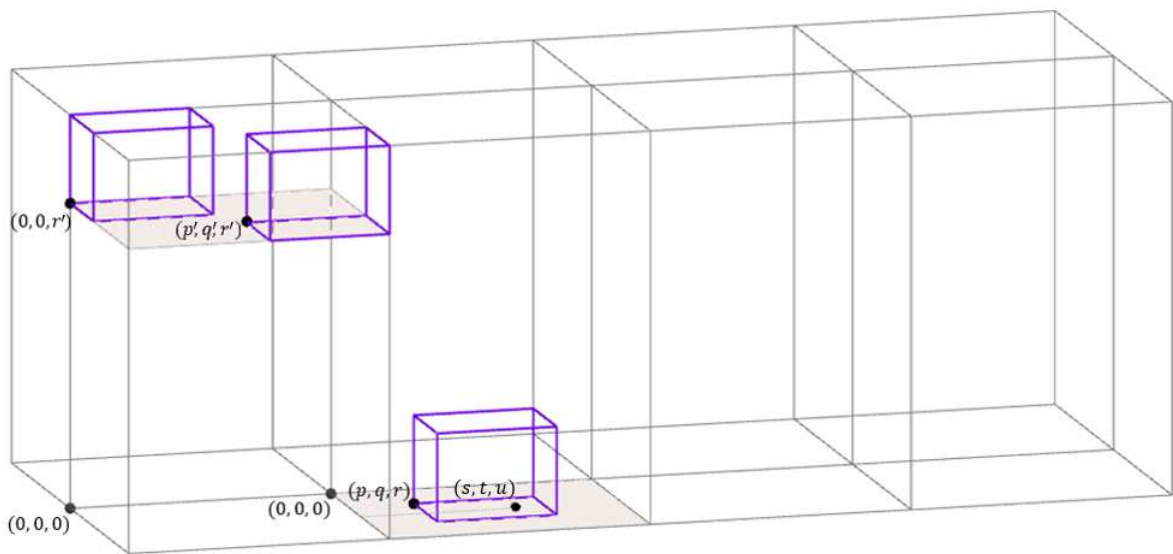
Conhecendo as posições disponíveis para o empacotamento e sabendo que os itens são distribuídos em camadas horizontais, é preciso verificar as possíveis posições que as camadas podem ser acomodadas nos compartimentos.

Uma camada completa é composta por itens de um mesmo tipo, logo, uma camada é dita do tipo i se formada por itens do tipo $i \in M$. Os itens são dispostos seguindo orientações que permitam acomodar a maior quantidade de itens possível na estrutura. Ranck Jr. (2014) considerou o valor máximo como um dado conhecido, assim, não determinando a posição dos itens na camada completa.

As dimensões da camada são consideradas com os mesmos comprimento e largura do compartimento (bem como as divisórias entre as camadas), tendo sua altura definida pela tipo de item que compõe a camada. Com isso, as camadas completas são acomodadas em posições a partir de um ponto de referência $(0, 0, r')$ com $r' \in Z_{ic}^0$.

Deste modo, as decisões a serem tomadas para as camadas completas se resumem em determinar em qual compartimento e que posição a camada deve ser colocada. Os itens que não possuem quantidade suficiente para montar uma camada completa são designados como itens residuais $i \in M'$ e destinados a compor as camadas incompletas.

Figura 3.9: Ilustração de itens acomodados em camadas completas e incompletas.



Fonte: Adaptado de Ranck Jr. (2014).

Na Figura 3.9 é possível observar três itens, de dimensões (l_i, w_i, h_i) , tendo dois item $i \in M$ alocados em uma camada completa a partir do ponto de referência $(0, 0, r')$ e outro item $i \in M'$ isolado que irá compor uma camada incompleta, na posição (p, q, r) .

3.2.2 Restrições Básicas e de Demanda

Durante o empacotamento dos itens e o processo de entrega, algumas restrições práticas são consideradas e precisam ser satisfeitas para a viabilidade do padrão de empacotamento. Primeiramente, as restrições de geometria buscam garantir que os itens não se sobreponham e estejam completamente acomodados no contêiner, juntamente com as restrições de atendimento de demanda, para garantir que todos os itens solicitados sejam carregados no contêiner.

Para estas restrições, consideramos os seguintes parâmetros e variáveis:

- C : conjunto dos compartimentos do contêiner;
- K : conjunto de destinos das entregas;
- i, j : índices dos tipos de itens ou camadas;
- s, t, u : índices das coordenadas em relação aos eixos x, y e z , respectivamente;
- d_{ik} : demanda do item de tipo i no destino k , $i \in M$, $k \in K$;
- Q_i^{max} : quantidade máxima de itens do tipo i que podem ser acomodados em uma camada;

- x_{jkpqr} : variável binária que é igual a 1 se o item do tipo $j \in \bar{M}$ e do destino $k \in K$, é colocado na posição (p, q, r) , com $p \in X_j^0$, $q \in Y_j^0$ e $r \in Z_{jc}^0$, do compartimento $c \in C$; e caso contrário, é igual a 0;
- \hat{x}_{ikrc} : variável que descreve a quantidade de itens do destino $k \in K$ em uma camada completa do tipo $i \in M$ colocada na posição $(0, 0, r)$, com $r \in Z_{ic}^0$, no compartimento $c \in C$;
- \hat{x}_{irc}^a : variável binária de decisão que é igual a 1 se há itens em uma camada do tipo $i \in M$, colocada na posição $(0, 0, r)$, com $r \in Z_{ic}^0$, no compartimento $c \in C$; caso contrário, é igual a 0.

$$\sum_{\substack{j \in \bar{M} \\ \text{se } f_{i,j}^{id}}} \sum_{c \in C} \sum_{p \in X_j^0} \sum_{q \in Y_j^0} \sum_{r \in Z_{jc}^0} x_{jkpqr} + \sum_{c \in C} \sum_{r \in Z_{ic}^0} \hat{x}_{ikrc} = d_{ik}, \quad \begin{matrix} i \in M \\ k \in K \end{matrix} \quad (3.5)$$

$$\sum_{j \in \bar{M}} \sum_{\substack{p \in X_j^0 \\ s-l_i < p \leq s}} \sum_{\substack{q \in Y_j^0 \\ t-w_i < q \leq t}} \sum_{\substack{r \in Z_{jc}^0 \\ u-h_i < r \leq u}} \sum_{k \in K} x_{jkpqr} + \sum_{i \in M} \sum_{r \in Z_{ic}^0} \hat{x}_{irc}^a \leq 1, \quad \begin{matrix} s \in X^0, t \in Y^0 \\ u \in Z_c^0, c \in C \end{matrix} \quad (3.6)$$

$$\sum_{k \in K} \hat{x}_{ikrc} \leq \hat{x}_{irc}^a Q_i^{max}, \quad \begin{matrix} i \in M \\ r \in Z_{ic}^0 \\ c \in C \end{matrix} \quad (3.7)$$

$$x_{jkpqr} \in \{0, 1\}, \quad \begin{matrix} j \in \bar{M}, k \in K \\ p \in X_j^0, q \in Y_j^0 \\ r \in Z_{ic}^0, c \in C \end{matrix} \quad (3.8)$$

$$\hat{x}_{ikrc} \in \mathbb{N}, \quad \begin{matrix} i \in M, k \in K \\ r \in Z_{ic}^0, c \in C \end{matrix} \quad (3.9)$$

$$\hat{x}_{irc}^a \in \{0, 1\}, \quad \begin{matrix} i \in M \\ r \in Z_{ic}^0 \\ c \in C \end{matrix} \quad (3.10)$$

As restrições (3.5) determinam que a demanda dos itens de cada cliente k seja completamente atendida, calculada pela quantidade total de itens do tipo $i \in M$ do destino k alocados nas camadas incompletas, seja o item na orientação original ou rotacionado (dada pela primeira parcela da restrição), somado com a quantidade total de itens do tipo $i \in M$ do destino k , alocados nas camadas completas (dada pela segunda parcela da restrição).

As restrições (3.6) determinam que qualquer posição (s, t, u) de qualquer compartimento $c \in C$ só pode ser ocupada por um item ou camada, garantindo que não haja sobreposição de itens e/ou camadas. É exemplificado na Figura 3.8 um ponto de referência de coordenadas (s, t, u) contido em um item.

As restrições (3.7) verificam que a quantidade de itens em uma camada completa do tipo $i \in M$, carregada na posição $(0, 0, r')$ no compartimento c , não ultrapasse a

quantidade máxima de itens que a camada pode acomodar. As restrições (3.8), (3.9) e (3.10) definem o domínio das variáveis de decisão.

3.2.3 Restrições de Estabilidade

Em seguida, apresentam-se as restrições associadas à estabilidade dos itens no padrão de carregamento, observando os sentidos vertical e horizontal. Como descrito em Ranck Jr. (2014), estas restrições estão associadas às restrições de múltiplos destinos e procuram garantir o equilíbrio dos itens a fim de evitar danificações pela movimentação nos compartimentos.

Os parâmetros e variáveis considerados nas restrições de estabilidade são:

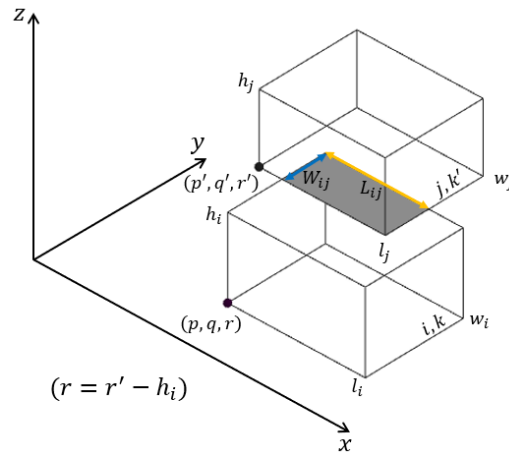
- ψ_i : quantidade mínima de itens que devem compor uma camada completa do tipo i para garantir estabilidade vertical necessária aos itens alocados acima dela;
- α : parâmetro de estabilidade vertical $\alpha \in \mathbb{R}$, tal que $0 \leq \alpha \leq 1$. Para $\alpha = 1$, é determinado que a face inferior de qualquer item esteja 100% apoiada em outros itens, camadas ou na base do compartimento. Quando $\alpha = 0$, não há exigência de estabilidade vertical para os itens;
- β : parâmetro de estabilidade horizontal para o eixo x , $\beta \in \mathbb{R}$, tal que $0 \leq \beta \leq 1$. Para $\beta = 1$, é necessário que a face lateral de qualquer item esteja 100% apoiada em outros itens, ou na face esquerda do compartimento. Para $\beta = 0$, indica-se que não há exigência de estabilidade horizontal no eixo x para os itens;
- γ : parâmetro de estabilidade horizontal para o eixo y , $\gamma \in \mathbb{R}$, tal que $0 \leq \gamma \leq 1$. Para $\gamma = 1$, é necessário que a face frontal de qualquer item esteja 100% apoiada em outros itens, ou na face da frente do compartimento. Para $\gamma = 0$, indica-se que não há exigência de estabilidade horizontal no eixo y para os itens.
- \mathcal{M} : número suficientemente grande;
- \hat{x}_{krc}^b : variável binária de decisão para verificar se uma camada foi alocada na posição $(0, 0, r)$, com $r \in Z_c^0$, no compartimento $c \in C$ e com itens demandados na parada $k \in K$, então é igual a 1; caso contrário, é igual a 0.

Estabilidade vertical

As restrições de estabilidade para o caso vertical buscam garantir que os itens estejam estáveis verticalmente ao longo de todo o percurso de entrega. Estas restrições requerem que os itens das camadas incompletas $j \in \bar{M}$ tenham uma fração mínima da sua face inferior apoiada na face superior de um ou mais itens, de destinos não anteriores ao destino desse item residual i , ou ainda, em uma camada completa, desde que esta possua uma quantidade mínima de itens (ψ_i) , de destinos não anteriores ao destino desse item j .

Em relação as camadas completas $j \in M$, estas devem estar apoiadas na base do compartimento ou em outras camadas completas, desde que esta possua uma quantidade mínima de itens (ψ_i), de destinos não anteriores ao destino dessa camada j .

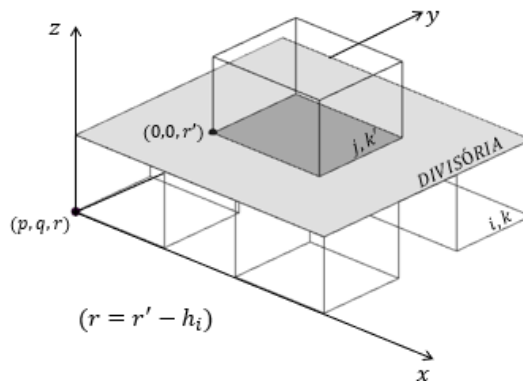
Figura 3.10: Estabilidade para o caso de um item apoiado em outro item.



Fonte: Adaptado de Ranck Jr. (2014).

Na Figura 3.10 é possível observar um exemplo do caso de estabilidade vertical entre dois itens, destacando a área de contato entre os itens, definida por $L_{ij} \cdot W_{ij}$. Um item $j \in \bar{M}$, com destino k' , tem sua face inferior apoiada em um item $i \in \bar{M}$, de destino $k \geq k'$.

Figura 3.11: Estabilidade para o caso de um item apoiado em uma camada.

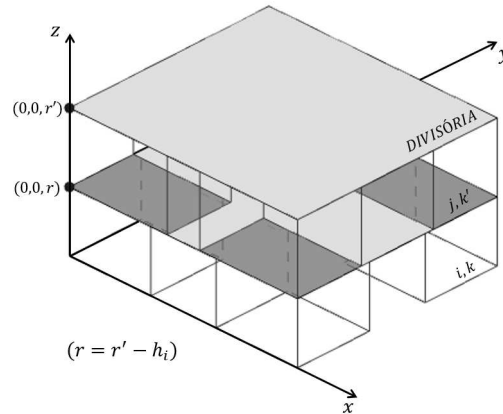


Fonte: Adaptado de Ranck Jr. (2014).

Na Figura 3.11 é possível constatar um exemplo para o caso de estabilidade vertical entre um item e uma camada. Um item $j \in \bar{M}$, de destino k' tem sua face inferior apoiada sobre uma camada de itens do tipo $i \in M$ com destino $k \geq k'$. A camada completa i sendo composta por 4 itens, suponto $\psi_i = 4$, a estabilidade vertical necessária para o item j é satisfeita.

Na Figura 3.12 é possível analisar um exemplo para o caso de estabilidade vertical entre duas camadas completas. Uma camada $j \in M$, com seus itens tendo como destino a parada k' , está apoiada na face superior de uma camada $i \in M$, composta por 4 itens

Figura 3.12: Estabilidade para o caso de uma camada apoiada em outra camada.



Fonte: Adaptado de Ranck Jr. (2014).

de destino $k \geq k'$. Considerando $\psi_i = 4$, como a camada completa i é composta de 4 itens, então a estabilidade vertical está satisfeita para a camada j .

Uma observação acerca destas restrições precisa ser considerada, em que, ao descarregar uma camada completa $j \in M$ de um compartimento $c \in C$, pode ocorrer a formação de um "buraco" no padrão de empacotamento. Para contornar esta situação, como já admitido, os itens podem ser movimentados apenas em relação ao eixo z . Com isso, é possível considerar que todos os itens deste compartimento c , acomodados em posições acima desta camada j podem ser movimentados para baixo, até que sejam apoiados por outras camadas ou pela base do compartimento.

As restrições são dadas por:

$$\sum_{\substack{i \in \bar{M} \\ r' - h_i \geq 0}} \sum_{\substack{k \in K \\ k \geq k'}} \sum_{\substack{p \in X_i^0 \\ p < p' + l_j \\ p > p' - l_i}} \sum_{\substack{q \in Y_i^0 \\ q < q' + w_j \\ q > q' - w_i}} L_{ij} W_{ij} x_{ikpq(r' - h_i)c} + \alpha l_j w_j \sum_{\substack{i \in M \\ r' - h_i \geq 0}} \sum_{k \in K} \hat{x}_{ik(r' - h_i)c} \geq \alpha l_j w_j \cdot x_{jk'p'q'r'c}, \quad \begin{array}{l} j \in \bar{M}, k' \in K \\ p' \in X_j^0, q' \in Y_j^0 \\ r' \in Z_c^0 / \{0\}, c \in C \end{array} \quad (3.11)$$

$$\sum_{\substack{i \in M \\ r' - h_i \geq 0}} \sum_{k \in K} \hat{x}_{ik(r' - h_i)c} \geq \sum_{\substack{j \in M \\ r' \leq H_c^{comp} - h_j}} \hat{x}_{jr'c}^a, \quad \begin{array}{l} r' \in Z_c^0 / \{0\} \\ c \in C \end{array} \quad (3.12)$$

$$\sum_{\substack{i \in M \\ r' \leq H_c^{comp} - h_i}} \sum_{\substack{k \in K \\ k \geq k'}} \left(\frac{1}{\psi_i} \right) \hat{x}_{ikr'c} + (1 - \hat{x}_{k'r'c}^b) \geq \hat{x}_{k'r'c}^b + \sum_{\substack{j \in \bar{M} \\ r' \leq H_c^{comp} - h_j \\ q \leq W_c^{comp} - w_j \\ p \leq L_c^{comp} - l_j}} \sum_{\substack{k \in K \\ k \geq k'}} x_{jkpqr'c}, \quad \begin{array}{l} k' \in K \\ p \in X^0, q \in Y^0 \\ r' \in Z_c^0 | r' < r \\ r \in Z_c^0, c \in C \end{array} \quad (3.13)$$

considerando:

$$L_{ij} = \min (p + l_i, p' + l_j) - \max (p, p')$$

$$W_{ij} = \min (q + w_i, q' + w_j) - \max (q, q')$$

As restrições (3.11) determinam que um item $j \in \bar{M}$ com destino $k' \in K$, acomodado em uma camada incompleta no compartimento c (verificado por $x_{jk'p'q'r'c} = 1$) tenha o suporte necessário para garantir a estabilidade vertical. Para o caso de item sobre item (Figura 3.9), verificado pela primeira parcela da restrição, a área entre a base do item j e a face superior do item i deve ser maior, ou igual, a uma parcela da área da base do item j , determinada por $\alpha \cdot l_j \cdot w_j$. Para o caso de item sobre uma camada (Figura 3.10), verificada pela segunda parcela da restrição, pode haver zero ou no mínimo ψ_i itens, quantidade mínima de itens em com camada completa, garantida pelas restrições (3.13).

Vale ressaltar que, apenas uma das parcelas nas restrições (3.11) será considerada no cálculo, uma vez que os dois casos não podem ocorrer simultaneamente, devido a restrição de não sobreposição, determinando que o item $j \in \bar{M}$ não pode estar apoiado sobre um item e também sobre uma camada completa. Ademais, o conjunto Z_{jc}^0 é considerado sem o zero, pois nesse caso, o suporte vertical para o item j é garantido pela base do compartimento.

Nas restrições (3.12), se algum $\hat{x}_{jr'c}^a = 1$, então existe uma camada completa do tipo $j \in M$ na posição $(0, 0, r')$ do compartimento. Neste caso, $\hat{x}_{ik(r'-h_i)c} = 1$, o que significa que deve existir uma camada completa de algum tipo i na posição $(0, 0, r' - h_i)$. Em relação ao conjunto Z_c^0 , não é considerado o elemento zero, tendo vista que neste caso, o suporte vertical garantido pela base do compartimento c .

As restrições (3.13) identificam a existência de algum item de destino $k' \in K$ acomodado em uma camada completa na posição $(0, 0, r)$, se $\hat{x}_{k'r c}^b = 1$, ou em uma camada incompleta, na posição (p, q, r) , se o somatório a esquerda for igual a 1, no compartimento c . Neste caso, as restrições asseguram que haja uma camada completa na posição $(0, 0, r')$, com $r' < r$, com zero itens ($\hat{x}_{k'r c}^b = 0$), ou no mínimo ψ_i itens, a fim de garantir o suporte vertical necessário, descrito por $\sum_{i \in M} \sum_{\{k \in K | k \geq k'\}} \hat{x}_{ikrc} \geq \psi_i$.

Vale ressaltar que, para não ocorrer a sobreposição de itens, o maior valor que o lado direito da desigualdade pode assumir é 1, garantindo que apenas um item ou uma camada completa ocupem a posição (p, q, r) .

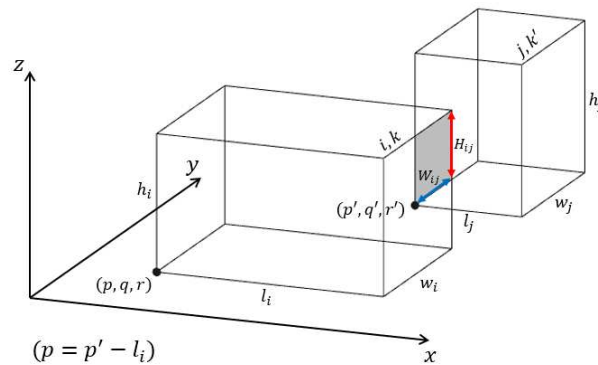
Estabilidade Horizontal

A restrição de estabilidade em relação a horizontal busca garantir que os itens estejam estáveis horizontalmente ao longo da rota de entrega. Para isso, é necessário analisar a situação em relação a ambos os eixos x e y . Como definido anteriormente (ver Seção 3.1.1), as camadas completas são tidas com o mesmo comprimento e largura do compartimento. Com isso, todas são consideradas horizontalmente estáveis, não sendo necessário definir restrições a elas, apenas aos itens que compõem as camadas incompletas.

Observando o posicionamento dos itens em relação ao eixo x , é necessário que um item $j \in \bar{M}$, tenha uma fração mínima da face lateral esquerda apoiada na lateral de um ou mais itens $i \in \bar{M}$, de paradas não anteriores a parada desse item j , ou na lateral do compartimento, garantindo o apoio lateral necessário. Considerando a posição dos itens em relação ao eixo y , é preciso que um item $j \in \bar{M}$, tenha uma fração mínima da face do fundo apoiada na frente de um ou mais itens $i \in \bar{M}$, de destinos não antecedentes ao destino desse item j , gerando a estabilidade necessária.

Na Figura 3.13 é possível observar uma ilustração para o caso de estabilidade horizontal em relação ao eixo x , na qual um item $j \in \bar{M}$ com destino $k' \in K$, tem sua face lateral esquerda apoiada na lateral direita de um item i de destino $k \in K$, com $k \geq k'$, com uma área de contato definida por $W_{ij} \cdot H_{ij}$.

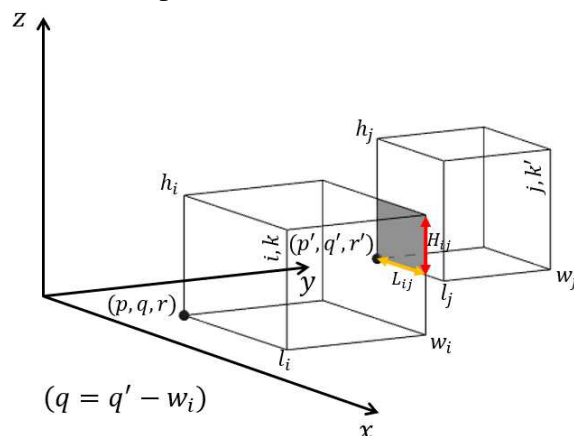
Figura 3.13: Exemplo de estabilidade horizontal no eixo x .



Fonte: Adaptado de Ranck Jr. (2014).

Na Figura 3.14 é possível observar uma ilustração para o caso de estabilidade horizontal em relação ao eixo y , na qual um item $j \in \bar{M}$ com destino $k' \in K$, tem sua face do fundo apoiada na frente de um item i de destino $k \in K$, com $k \geq k'$, com uma área de contato definida por $L_{ij} \cdot H_{ij}$.

Figura 3.14: Exemplo de estabilidade horizontal no eixo y .



Fonte: Adaptado de Ranck Jr. (2014).

As restrições são dadas por:

$$\sum_{\substack{i \in \bar{M} \\ p' - l_i \geq 0}} \sum_{\substack{k \in K \\ k \geq k'}} \sum_{\substack{q \in Y_i^0 \\ q < q' + w_j \\ q > q' - w_i}} \sum_{\substack{r \in Z_{ic}^0 \\ r < r' + h_j \\ r > r' - h_i}} W_{ij} H_{ij} x_{ik(p'-l_i)qrc} \geq \beta w_j h_j x_{jk'p'q'r'c}, \quad \begin{array}{l} j \in \bar{M} \\ p' \in X_j^0 / \{0\} \\ q' \in Y_j^0 \\ r' \in Z_{jc}^0 \\ k' \in K \\ c \in C \end{array} \quad (3.14)$$

$$\sum_{\substack{i \in \bar{M} \\ q' - w_i \geq 0}} \sum_{\substack{k \in K \\ k \geq k'}} \sum_{\substack{p \in X_i^0 \\ p < p' + l_j \\ p > p' - l_i}} \sum_{\substack{r \in Z_{ic}^0 \\ r < r' + h_j \\ r > r' - h_i}} L_{ij} H_{ij} x_{ikp(q'-w_i)rc} \geq \gamma l_j h_j x_{jk'p'q'r'c} \quad \begin{array}{l} j \in \bar{M} \\ p' \in X_j^0 \\ q' \in Y_j^0 / \{0\} \\ r' \in Z_{jc}^0 \\ k' \in K \\ c \in C \end{array} \quad (3.15)$$

considerando

$$\begin{aligned} L_{ij} &= \min(p + l_i, p' + l_j) - \max(p, p') \\ W_{ij} &= \min(q + w_i, q' + w_j) - \max(q, q') \\ H_{ij} &= \min(r + h_i, r' + h_j) - \max(r, r') \end{aligned}$$

As restrições (3.14) determinam que um item $j \in \bar{M}$ com destino $k' \in K$, acomodado em uma camada incompleta no compartimento c (verificado por $x_{jk'p'q'r'c} = 1$) tenha o suporte necessário para garantir a estabilidade horizontal em relação ao eixo x (Figura 3.12). Para este caso, a área de contato entre a face lateral esquerda do item j e a face lateral direita de um de item i , ou um conjunto de itens, deve ser igual, ou maior, a uma parcela da área da face lateral do item j , determinada por $\beta \cdot w_j \cdot h_j$.

As restrições (3.15) estipulam que um item $j \in \bar{M}$ de destino $k' \in K$, alocado em uma camada incompleta no compartimento c (verificado por $x_{jk'p'q'r'c} = 1$) tenha o suporte necessário para garantir a estabilidade horizontal em relação ao eixo y (Figura 3.13). Para este caso, a área de contato entre a face da frente do item j e as faces do fundo de outros itens $i \in \bar{M}$ deve ser igual, ou maior, a uma parcela da área da face posterior do item j , calculada por $\gamma \cdot l_j \cdot h_j$.

3.2.4 Restrições de Resistência ao Empilhamento

Conforme apresentam Junqueira (2009) e Ranck Jr. (2014), consideram-se dois casos para a resistência ao empilhamento: itens acomodados em camadas completas e em camadas incompletas. Assim como propõem os autores, considera-se que:

- os itens precisam ter suas bases 100% apoiadas na face superior de outros itens, camadas ou na base do compartimento (identificado nas restrições por $\alpha = 1$);
- a posição dos itens no contêiner, no plano xy , não são modificadas ao longo da rota de entrega;

- c) cada ponto da face superior de um item sustenta a mesma pressão considerada para o item, da mesma forma que a pressão exercida por um item é igualmente aplicada em cada ponto da sua base.

Os parâmetros considerados nestas restrições são dados por:

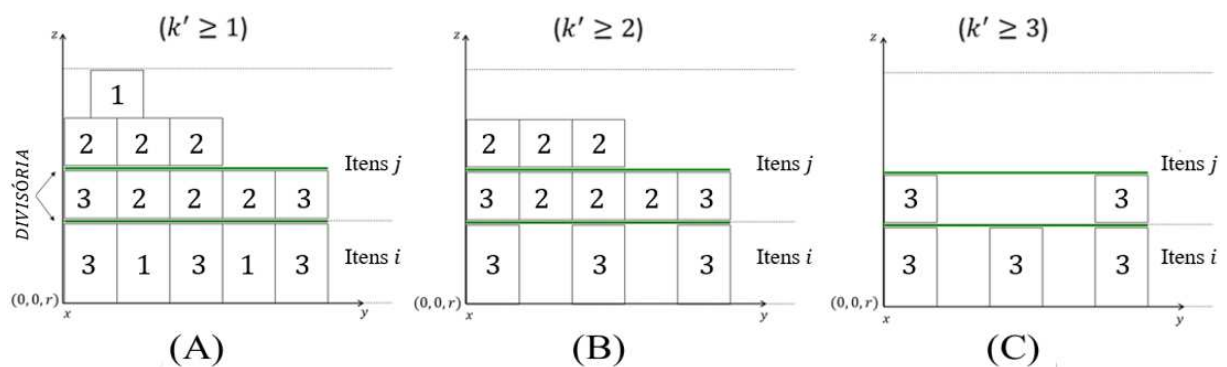
- P_i : peso de um item do tipo i ;
- σ_i : pressão máxima suportada em qualquer ponto da face superior de um item do tipo i .

Camadas Completas

As restrições de empilhamento para o caso de itens acomodados em camadas completas, considera-se que o peso exercido sobre uma camada é distribuído sobre todos os itens que compõem esta camada. Este conjunto de restrições deve ser satisfeito para cada parada $k \in K$, considerando o carregamento com os itens a serem entregues na parada k e nas paradas seguintes.

Em um primeiro caso, a pressão exercida sobre uma camada completa pode variar conforme as entregas são realizadas, em que a quantidade de itens na camada e sobre ela vai diminuindo. Na Figura 3.15 é possível observar uma situação de entrega de itens dos tipos i e j , de um carregamento composto pela demanda de três paradas ($k = 1, 2$ e 3). Identifica-se como $k' \geq 1$, quando estão acomodados no contêiner, os itens destinados para parada 1 e para as paradas seguintes. Além disso, consideramos a quantidade mínima de itens em uma camada completa igual a 3 ($\psi_i = \psi_j = 3$).

Figura 3.15: Empilhamento de itens de diferentes destinos.



Fonte: Ranck Jr. (2014).

Na Figura 3.15 (A), enquanto nenhum item foi entregue ($k' \geq 1$), a pressão exercida a camada i , posicionada na posição $(0,0,r)$, é dada pelo peso aplicado pelos itens acomodados sobre ela, calculado por $9 \cdot P_j$, dividido pela área da face superior da camada completa, ou seja, a soma das faces superiores dos itens que compõem a camada completa do tipo i , calculada por $5 \cdot w_i \cdot l_i \cdot \sigma_i$. Da maneira análoga, na Figura 3.15 (B), a pressão exercida

sobre a camada i , quando ($k' \geq 2$), é calculado por $8 \cdot P_j / 3 \cdot w_i \cdot l_i \cdot \sigma_i$. Por fim, na Figura 3.15 (C), quando apenas os itens da última parada estão no contêiner, a pressão exercida sobre a camada i é calculada por $2 \cdot P_j / 3 \cdot w_i \cdot l_i \cdot \sigma_i$.

As restrições são dadas por:

$$\begin{aligned} & \sum_{j \in \bar{M}} \sum_{\substack{k' \in K \\ k' \geq k}} \sum_{p' \in X_j^0} \sum_{q' \in Y_j^0} \sum_{\substack{r' \in Z_{jc}^0 \\ r' > r}} P_j x_{jk'p'q'r'c} + \sum_{j \in M} \sum_{\substack{k' \in K \\ k' \geq k}} \sum_{\substack{r' \in Z_{jc}^0 \\ r' > r}} P_j x_{jk'r'c} \\ & \leq \sum_{i \in M} \sum_{\substack{k' \in K \\ k' \geq k}} w_i l_i \sigma_i \hat{x}_{ik'r'c} + \mathcal{M}(1 - \hat{x}_{krc}^b), \quad \begin{matrix} r \in Z_c^0 \\ k \in K \\ c \in C \end{matrix} \end{aligned} \quad (3.16)$$

As restrições (3.16) determinam, para uma camada alocada na posição $(0, 0, r)$ do compartimento c , composta por itens da parada $k \in K$ ou paradas seguintes ($\hat{x}_{krc}^b = 1$), que o peso total dos itens acomodados acima dela, calculada pelos fatores do lado esquerdo da desigualdade, não supere o fator de resistência à pressão desta camada, calculado a partir da área total das faces superiores dos itens que montam a camada, multiplicada pela pressão máxima sustentada por ela em cada ponto da sua face superior. Caso contrário ($\hat{x}_{krc}^b = 0$), a restrição é redundante, uma vez que o valor resultante do lado direito da desigualdade será \mathcal{M} .

Camadas Incompletas

As restrições de empilhamento para o caso de itens acomodados em camadas incompletas, considera-se que o peso exercido sobre um item é distribuído igualmente em todos os pontos da face superior deste item. Destaca-se que para estas camadas, a pressão exercida sobre os itens não aumenta no decorrer do percurso de entrega (RANCK Jr., 2014), uma vez que as caixas vão sendo retiradas do contêiner em cada parada. Com isso, as restrições para camadas incompletas podem ser verificadas apenas na primeira parada ($k = 1$), considerando o momento em que todos os itens estão carregados no contêiner e nenhuma entrega ainda tenha sido realizada ($k' \geq 1$).

As restrições são dadas por:

$$\begin{aligned} & \sum_{j \in \bar{M}} \sum_{\substack{k' \in K \\ k' \geq k}} \sum_{\substack{p' \in X_j^0 \\ s-l_j < p' \leq s}} \sum_{\substack{q' \in Y_j^0 \\ t-w_j < q' \leq t}} \sum_{\substack{r' \in Z_{jc}^0 \\ r' > u}} \left(\frac{P_j}{l_j w_j} \right) x_{jk'p'q'r'c} \\ & \leq \sum_{i \in \bar{M}} \sum_{\substack{k' \in K \\ k' \geq k}} \sum_{\substack{p \in X_i^0 \\ s-l_i < p \leq s}} \sum_{\substack{q \in Y_i^0 \\ t-w_i < q \leq t}} \sum_{\substack{r \in Z_{ic}^0 \\ u-h_i < r \leq u}} \sigma_i x_{ik'pqr'c} \\ & + \mathcal{M} \cdot \left(1 - \sum_{i \in \bar{M}} \sum_{\substack{k' \in K \\ k' \geq k}} \sum_{\substack{p \in X_i^0 \\ s-l_i < p \leq s}} \sum_{\substack{q \in Y_i^0 \\ t-w_i < q \leq t}} \sum_{\substack{r \in Z_{ic}^0 \\ u-h_i < r \leq u}} x_{ik'pqr'c} \right), \quad \begin{matrix} s \in X^0 \\ t \in Y^0 \\ u \in Z_c^0 \\ k=1 \\ c \in C \end{matrix} \end{aligned} \quad (3.17)$$

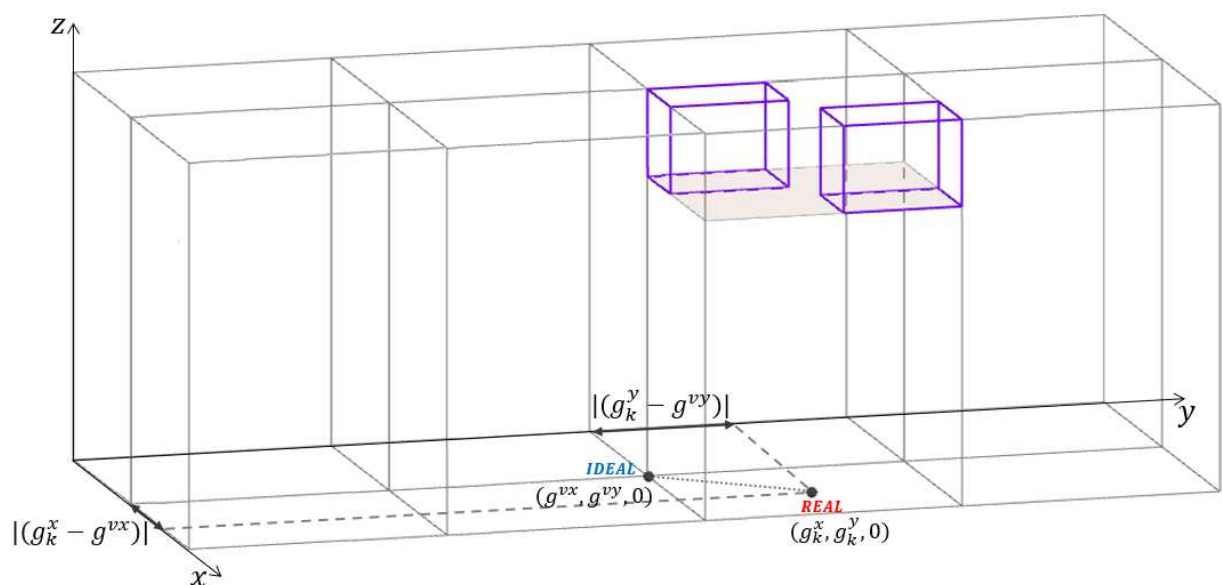
As restrições (3.17) determinam, para algum item $i \in \bar{M}$ com destino a parada $k' \geq k$, acomodado em uma camada incompleta na posição (p, q, r) do compartimento c e ocupando uma posição determinada (s, t, u) deste compartimento, que o peso total dos itens j , com destinos iguais ou posteriores ao destino do item i e acomodados acima desta posição ($x_{ik'pqrc} = 1$), não supere o fator de resistência à pressão (σ_i) deste item. Caso contrário ($x_{ik'pqrc} = 0$), a restrição é redundante, uma vez que o valor resultante do lado direito da desigualdade será \mathcal{M} .

3.2.5 Balanceamento de Peso

No processo de carregamento, a distribuição eficiente dos itens é necessária para garantir um bom balanceamento do peso da carga no contêiner, buscando um transporte sem danos dos produtos e também, cuidados preventivos no desgaste do veículo. Conforme apresentado no Capítulo 3, Seção 3.1, no problema analisado, as restrições de balanceamento são consideradas fracas, sendo assim, penalizadas na função objetivo (RANCK Jr., 2014).

Definindo as possíveis posições para os itens no padrão de carregamento, a fim de evitar a formulação das restrições e das funções objetivos não lineares, os desvios de balanceamento são verificados em relação aos eixos x e y , separadamente. Os desvios são calculados a partir da diferença entre o centro de gravidade ideal, considerado usualmente no centro geométrico no piso do contêiner, e o real centro de gravidade dos itens carregados. A formulação adotada para estas restrições é calculada para cada parada $k \in K$, a partir do somatória das diferenças dos centros de gravidades dos itens e das camadas acomodados no contêiner, demandados pela parada k .

Figura 3.16: Centros de gravidade para cálculo dos desvios de balanceamento.



Fonte: Adaptado de Ranck Jr. (2014).

Na Figura 3.16 é possível verificar as possíveis posições para os centros de

gravidade ideal e real para o cálculos dos desvios de balanceamento, considerando itens da parada $k \in K$, com seus centro de gravidade projetado no piso do compartimento em que foram carregados.

Os parâmetros e variáveis considerados nestas restrições são:

- g^{vx}, g^{vy} : centro de gravidade ideal para os itens acomodados no contêiner em relação aos eixos x e y , definido a partir do sistema de coordenadas do contêiner (ver Seção 3.1.1);
- P_k^Δ : peso total da carga quando apenas os itens destinados a parada k e posteriores ($k' \geq k$) estão no contêiner;
- $\varepsilon^x, \varepsilon^y$: coeficientes de tolerância para os centros de gravidades, sendo $\varepsilon^x \geq 0$ e $\varepsilon^y \geq 0$;
- τ_i^x, τ_i^y : centro de gravidade do item $i \in \bar{M}$ em relação aos eixos x e y , respectivamente, definidos de acordo com o sistema de coordenadas do item;
- ω^{dev} : peso relativo aos desvios de balanceamento de peso no contêiner;
- g_k^x, g_k^y : variáveis auxiliares que definem o centro de gravidade real dos itens demandados nas paradas k' , com $k' \geq k$, em relação aos eixos x e y , respectivamente, definido a partir do sistemas de coordenadas do contêiner;
- θ_k^x, θ_k^y : variáveis de decisão que determinam os desvios de balanceamento de peso no contêiner em relação aos eixos x e y , respectivamente, para os itens demandados nas paradas k' , com $k' \geq k$.

As restrições e a função objetivo consideradas para o problema de balanceamento de peso são:

$$Min \omega^{dev} \sum_{k \in K} (\theta_k^x + \theta_k^y) \quad (3.18)$$

$$g_k^x = \frac{\left(\sum_{i \in \bar{M}} \sum_{\substack{k' \in K \\ k' \geq k}} \sum_{c \in C} \sum_{p \in X_i^0} \sum_{q \in Y_i^0} \sum_{r \in Z_{ic}^0} P_i (p_c + p + \tau_i^x) x_{ik'pqr c} \right) + \sum_{i \in M} \sum_{\substack{k' \in K \\ k' \geq k}} \sum_{c \in C} \sum_{r \in Z_{ic}^0} P_i \left(p_c + \frac{L^{comp}}{2} \right) \hat{x}_{ik'rc}}{P_k^\Delta}, \quad k \in K \quad (3.19)$$

$$g_k^y = \frac{\left(\sum_{i \in \bar{M}} \sum_{\substack{k' \in K \\ k' \geq k}} \sum_{c \in C} \sum_{p \in X_i^0} \sum_{q \in Y_i^0} \sum_{r \in Z_{ic}^0} P_i(q_c + q + \tau_i^y) x_{ik'pqr} \right) + \sum_{i \in M} \sum_{\substack{k' \in K \\ k' \geq k}} \sum_{c \in C} \sum_{r \in Z_{ic}^0} P_i\left(q_c + \frac{w^{comp}}{2}\right) \hat{x}_{ik'rc}}{P_k^\Delta}, \quad k \in K \quad (3.20)$$

$$\begin{aligned} \theta_k^x &\geq P_k^\Delta (g^{vx} - g_k^x) - \varepsilon^x \\ \theta_k^x &\geq P_k^\Delta (g_k^x - g^{vx}) - \varepsilon^x, \end{aligned} \quad k \in K \quad (3.21)$$

$$\begin{aligned} \theta_k^y &\geq P_k^\Delta (g^{vy} - g_k^y) - \varepsilon^y \\ \theta_k^y &\geq P_k^\Delta (g_k^y - g^{vy}) - \varepsilon^y, \end{aligned} \quad k \in K \quad (3.22)$$

$$\theta_k^x, \theta_k^y \in \mathbb{R}, \quad \theta_k^x \geq 0, \quad \theta_k^y \geq 0, \quad k \in K \quad (3.23)$$

A função (3.18) determina o termo da função objetivo responsável pela minimização das penalizações dos desvios de balanceamento do peso da carga alocada no contêiner. As restrições (3.19) e (3.20) determinam o centro de gravidade real da carga, respectivamente para os eixos x e y , considerando o sistema de coordenadas definido pelo contêiner. A primeira parcela das restrições definem a posição relativa ao centro de gravidade de um item $i \in \bar{M}$, acomodado em uma camada incompleta. O centro de gravidade é calculado pela soma do centro de gravidade do item i , com sua posição de referência e a posição de referência do compartimento em relação ao eixo considerado, multiplicado pelo peso do item i . Na segunda parcela das restrições, é definida a posição relativa do centro de gravidade de uma camada completa $i \in M$, sendo calculada pela soma da posição de referência do compartimento com a metade da dimensão (largura ou comprimento) considerada na equação, multiplicada pelo peso do item i .

As restrições (3.21) e (3.22) determinam os desvios de balanceamento do peso da carga alocada no contêiner em relação aos eixos x e y , respectivamente, quando estão no contêiner apenas os itens dos destinos k e k' , com $k' \geq k$. Os desvios de balanceamento são calculados a partir da diferença entre os centros de gravidade ideal e real, multiplicada pelo peso total da carga alocada no contêiner na parada k . As restrições são consideradas apenas quando os desvios de balanceamento apresentam valores maiores que os valores de tolerância ε^x e ε^y .

As restrições (3.23) determinam o domínio das variáveis de decisão. Vale ressaltar que, as restrições para os desvios de balanceamento são válidas, apenas em associação a função objetivo (3.18). Ademais, "a fim de manter a formulação linear e sem perder a generalidade, é definida uma função módulo com duas desigualdades lineares" (RANCK Jr., 2014). É importante ressaltar que, considerando estas restrições juntamente com padrões normais (ver Seção 3.1.1), pode acarretar em perda de generalidade da solução, deixando de obter a solução ótima para o problema.

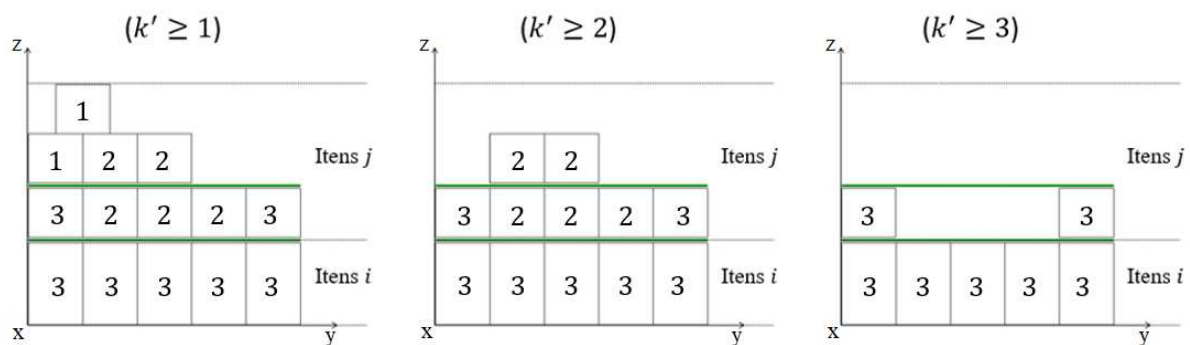
3.2.6 Remanejamento de Itens

O processo de remanejar itens durante a rota de entrega é empregado conforme a necessidade da movimentação de alguns itens para acessar aqueles que são demandados na parada que está sendo visitada. Como definido na, Seção 3.2, o remanejamento de produtos em uma parada $k \in K$ ocorre por descarregar alguns itens e acomodá-los temporariamente fora do contêiner, para assim, acessar os itens a serem entregues na parada k e logo após, alocar novamente os itens no contêiner.

Considerando as estruturas em que os itens são acomodados (camadas completas e incompletas), quando uma camada completa $i \in M$ é composta por itens de mais de um destino, ao descarregar alguns itens desta camada i , a estrutura pode deixar de ser completa. Caso a camada não atenda a quantidade mínima de itens necessários (ψ_i), é preciso que as caixas restantes da camada sejam remanejadas e acomodadas nas camadas incompletas. Em relação as camadas incompletas, compostas por diferentes tipos de itens, de diferentes destinos, e sendo necessário para estabilidade o apoio de 100% da base dos itens, estas estruturas devem estar sempre acomodadas sobre as camadas completas, na parte mais alta do compartimento, e com isso, os itens precisam ser descarregados sempre que necessário acessar uma camada completa abaixo.

Na Figura 3.17 é possível observar um exemplo de carregamento com itens dos tipos i e j destinados as paradas $k' = 1, 2, 3$, em que não é necessário o remanejamento de itens para acessar os produtos demandados em cada parada k' durante a rota de entrega e a estabilidade dos itens não é comprometida.

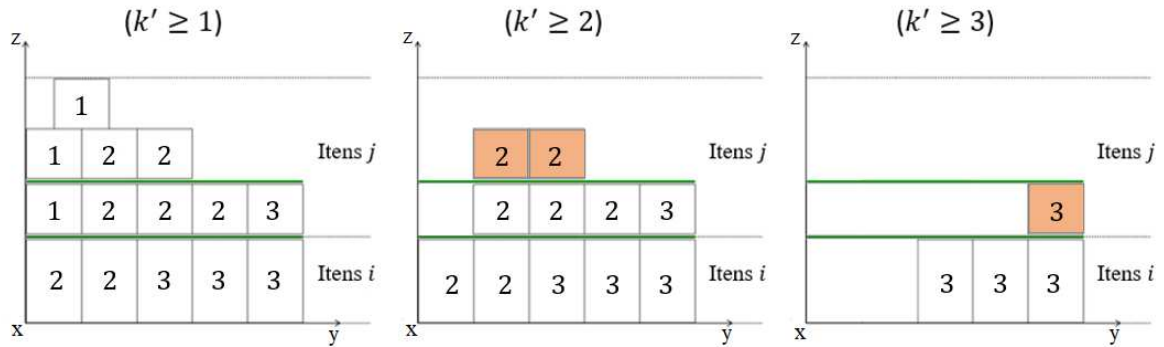
Figura 3.17: Carregamento de itens sem remanejamento necessário.



Fonte: Ranck Jr. (2014).

Na Figura 3.18 é possível observar uma ocasião em que o carregamento dos itens de tipos i e j destinados as paradas $k' = 1, 2, 3$, é necessário o remanejamento dos itens a fim de garantir a estabilidade da carga. Na parada $k' = 1$, é necessário remanejar um dos itens de tipo j destinados à parada $k' = 2$. Ao realizar a entrega na parada $k' = 2$, um item de tipo j destinado a parada $k' = 3$ precisa ser remanejado.

Figura 3.18: Carregamento de itens com remanejamento necessário.



Fonte: Ranck Jr. (2014).

Os parâmetros e variáveis considerados nas restrições e na função objetivo são:

- ω^z : peso relativo para o remanejamento de itens;
- ω^{zq}, ω^{zp} : pesos relativos para a quantidade de itens remanejados e o peso dos itens remanejados, respectivamente;
- \hat{x}_{krc}^d : variável binária de decisão, quando igual a 1, determina se uma camada completa alocada na posição $(0, 0, r)$, $r \in Z_{ic}^0$, no compartimento $c \in C$ contém itens da parada $k \in K$; caso contrário, é igual a 0;
- z_{jkc} : variável inteira de decisão que determina a quantidade de itens de um mesmo tipo j , acomodados em camadas completas ($j \in M$) e em camadas incompletas ($j \in \bar{M}$), a serem remanejados no compartimento $c \in C$ na parada $k \in K$.

As restrições e a função objetivo consideradas no remanejamento de itens são:

$$\text{Min } \omega^z \sum_{j \in M} \sum_{k \in K/\{|K|\}} \sum_{c \in C} (\omega^{zq} + \omega^{zp} P_j) z_{jkc} \quad (3.24)$$

$$\geq \sum_{\substack{k' \in K \\ k' > k}} \left(\sum_{\substack{r' \in Z_{jc}^0 \\ r' > r}} \hat{x}_{jk'r'c} + \sum_{\substack{j' \in \bar{M} \\ j' = j}} \sum_{\substack{r' \in Z_{j'c}^0 \\ r' > r}} \sum_{p' \in X_{j'}^0} \sum_{q' \in Y_{j'}^0} x_{j'k'p'q'r'c} \right), \quad \begin{array}{l} r \in Z_c^0 / \\ \{ \text{Max}_{r' \in Z_{c0}(r')} \} \\ k \in K/\{|K|\} \\ j \in M \\ c \in C \end{array} \quad (3.25)$$

$$\sum_{\substack{i \in M \\ r \leq H_c^{\text{comp}} - h_i}} \hat{x}_{ikrc} \leq \mathcal{M} \hat{x}_{krc}^d, \quad \begin{array}{l} r \in Z_c^0 / \\ \{ \text{Max}_{r' \in Z_{c0}(r')} \} \\ k \in K/\{|K|\} \\ c \in C \end{array} \quad (3.26)$$

$$\hat{x}_{krc}^d \in \{0, 1\}, \quad \begin{array}{l} r \in Z_c^0 / \\ \{ \text{max}_{r' \in Z_{c0}(r')} \} \\ k \in K/\{|K|\} \\ c \in C \end{array} \quad (3.27)$$

$$z_{jkc} \in \mathbb{N}, \quad \begin{array}{l} j \in M \\ k \in K/\{|K|\} \\ c \in C \end{array} \quad (3.28)$$

A função (3.24) determina o termo da função objetivo responsável por verificar as penalizações com o remanejamento dos itens durante toda percurso de entregas, considerando a quantidade e o peso dos itens remanejados em cada parada $k \in K/\{|K|\}$. Destaca-se que, para o remanejamento, é desconsiderada a última parada $k = |K|$ do conjunto de destinos K , uma vez que, neste momento, todos os demais itens já foram entregues e com isso, nenhum remanejamento será necessário.

As restrições (3.25) determinam quando é necessário o remanejamento de uma camada completa $j, j' \in \bar{M}$, sendo $j = j'$, em uma quantidade dada por z_{jkc} , em uma dada parada k . Os itens que precisam ser remanejados possuem destinos k' , sendo $k' > k$ e estão posicionados acima da camada completa, no caso em que $\hat{x}_{krc}^d = 1$. Caso contrário, se não há nenhuma camada completa acomodada em uma posição abaixo de itens da parada k' , sendo $k' > k$, então $\hat{x}_{krc}^d = 0$, tornando a expressão redundante, uma vez que o lado direito da desigualdade será igual a \mathcal{M} .

Nas restrições (3.26) é verificada a existência de ao menos um item com destino à parada k , fazendo parte de em uma camada completa acomodada na posição $(0, 0, r)$ no compartimento c , definida por $\hat{x}_{krc}^d = 1$. Caso contrário, $\hat{x}_{krc}^d = 0$.

As restrições (3.27) e (3.28) determinam os domínios das variáveis de decisão. É relevante ressaltar que a variável \hat{x}_{krc}^d não é definida para última parada da lista de destinos $k = |K|$, uma vez que esta variável é utilizada nas restrições (3.25) associada à variável z_{jkc} . Ademais, a variável \hat{x}_{krc}^d também não é definida para $r = \max_{r' \in Z_c^0}(r')$, que determina a posição mais alta que uma camada pode ocupar, ou seja, nenhum item é acomodado acima desta posição.

Com as restrições definidas, a função objetivo pode ser escrita como:

$$\text{Min } \omega^z \sum_{j \in M} \sum_{k \in K/\{|K|\}} \sum_{c \in C} (\omega^{zq} + \omega^{zp} P_j) z_{jkc} + \omega^{dev} \sum_{k \in K} (\theta_k^x + \theta_k^y) \quad (3.29)$$

Uma versão final da função objetivo pode ser estabelecida, tendo as camadas completas e incompletas definidas, assim é possível determinar a função objetivo considerando os tipos de itens demandas e ordem de paradas, reescrevendo a função (3.29) como:

$$\text{Min } \omega^z \sum_{k \in K/\{|K|\}} \sum_{j \in M} (\omega^{zq} Q_{jk+1} + \omega^{zp} P_{jk+1}) z_{jk} + \omega^{dev} \sum_{k \in K} (\theta_k^x + \theta_k^y) \quad (3.30)$$

Na função objetivo (3.30), a primeira parcela está relacionada as penalizações para o remanejamento de itens, considerando a quantidade de caixas (Q_{jk+1}) e o peso das caixas (P_{jk+1}) a serem remanejadas na parada $k \in K$. A segunda parcela está relacionada as penalizações para os desvios de balanceamento em relação aos eixos x e y , para os itens a serem entregues nas parada k' , com $k' \geq k$.

4 MÉTODO DE SOLUÇÃO PROPOSTO

Neste capítulo é apresentado o método de solução proposto, em que são retomados conceitos da definição do problema, explanados no Capítulo 3, Seção 3.1, e algumas observações acerca do processo de empacotamento e os parâmetros considerados. Na sequência, são apresentadas a etapa referente a construção das camadas completas, (Seção 4.1) e a etapa correspondente a construção das camadas incompletas (Seção 4.2).

A solução para este problema foi delineada inicialmente por Ranck Jr. (2014), obtida a partir da resolução de modelos de Programação Linear Inteira Mista. No entanto, é importante destacar que soluções geradas com essa abordagem são obtidas com um elevado tempo computacional. Com o intuito de contornar essa situação, o foco deste trabalho é desenvolver uma heurística construtiva com o objetivo de obter soluções factíveis e de valores condizentes aos apresentados por Ranck Jr. (2014), de forma mais eficiente em termos de tempo de resolução.

A heurística construtiva proposta é um procedimento determinístico que segue um conjunto de instruções para guiar o processo de solução. A proposta consiste em realizar o empacotamento dos itens, distribuindo-os em camadas horizontais, separadas por divisórias, nos compartimentos que compõem o contêiner, verificando as restrições consideradas na modelagem do problema (ver Seção 3.2). Visando a minimização do manuseio da carga durante o percurso de entrega, o carregamento segue as demandas da lista de clientes em ordem inversa, para que, os primeiros itens a serem carregados sejam os últimos a serem entregues.

O remanejamento de um item ocorre quando um item está acomodado sobre outros itens de destinos anteriores e que precisam ser descarregados para entrega. O balanceamento da carga no contêiner, como é considerado como uma restrição fraca, é abordado em segundo plano e penalizado na função objetivo.

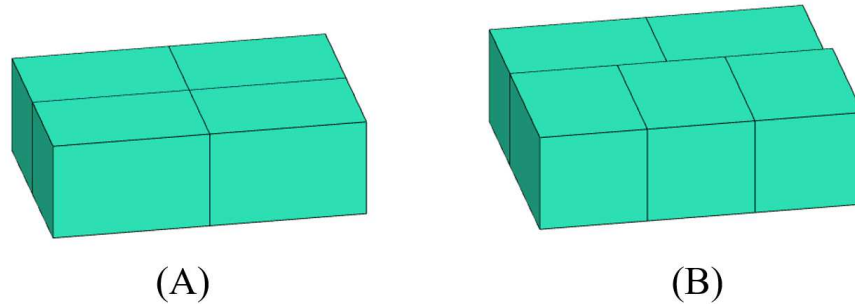
A seguir são retomados alguns conceitos adotados na construção e no carregamento das camadas completas e das camadas incompletas, definidos na Seção 3.1, algumas observações acerca do processo de empacotamento e os parâmetros considerados no método heurístico.

Camadas Completas

As camadas completas são estruturas horizontais compostas por itens de um mesmo tipo $i \in M$, acomodados seguindo a orientação inicial, ou com uma rotação de 90° no plano xy , de maneira que possibilite o arranjo do maior número de caixas na camada, sem que produtos sejam empilhados. Na Figura 4.1 (A) é possível observar um exemplo de camada completa que considera apenas uma orientação para o empacotamento dos itens e na Figura 4.1 (B), é possível analisar um exemplo de camada completa composta por itens empacotados em

duas orientações.

Figura 4.1: Exemplos de camadas completas.



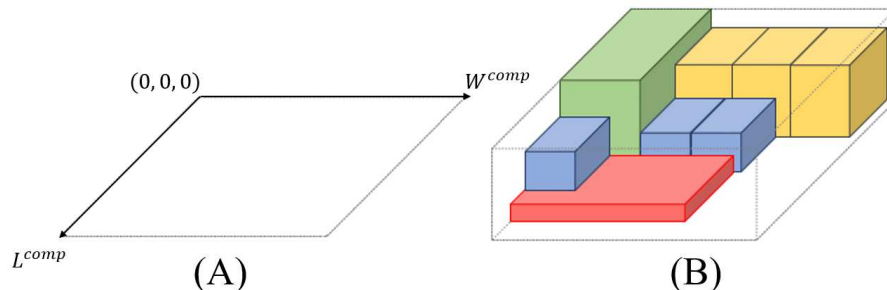
Fonte: o Autor.

Os itens que não fazem parte das camadas completas são denominados itens *residuais*, repassados para o conjunto de itens residuais (M') e destinados a compor as camadas incompletas.

Camadas Incompletas

As camadas incompletas podem ser compostas por diferentes tipos de itens $i \in M'$, os itens podem ser empacotados com orientação 0° e 90° em relação ao plano xy e a estrutura comporta o empilhamento de caixas. O padrão de empacotamento que define uma camada incompleta é estabelecido resolvendo um problema de carregamento tridimensional e, por isso, tem um sistema de coordenadas próprio, com origem definida em seu canto inferior posterior esquerdo. Na Figura 4.2 (A) é apresentado um exemplo de um sistema de coordenadas para uma camada incompleta e na Figura 4.2 (B) é apresentado um possível padrão de empacotamento para esta camada com diferentes tipos de itens.

Figura 4.2: Exemplo de uma camada incompleta.

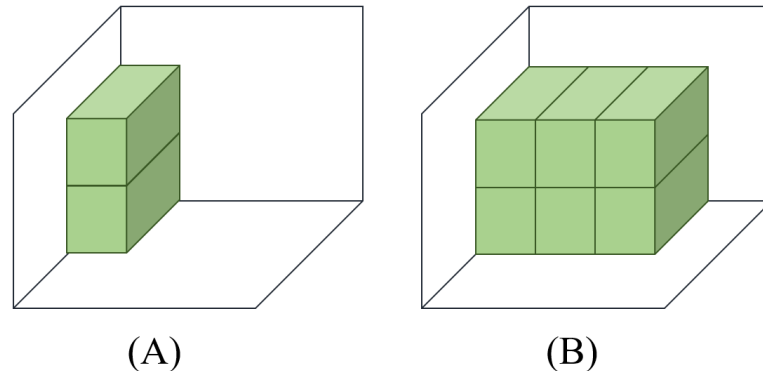


Fonte: Adaptado de Ranck Jr. (2014).

Para o método heurístico, estabelecemos a composição das camadas incompletas a partir da construção de paredes verticais. Acomodando o primeiro item na origem do espaço livre, inicialmente sendo a posição $(0, 0, 0)$ do sistema de coordenadas da camada incompleta. Em seguida, os itens são distribuídos em blocos. Os blocos são formados por pilhas

de itens, acomodando o máximo de caixas possível em relação a altura do espaço livre, como apresentado na Figura 4.3 (A). Logo após, novas pilhas são montadas ao lado da primeira estrutura, até completar a largura do espaço livre, como ilustrado na Figura 4.3 (B). As pilhas são compostas por itens de um mesmo tipo $i \in M'$ e construídas considerando as demandas de uma mesma parada $k \in K$, seguindo a ordem inversa de entregas.

Figura 4.3: Exemplo de distribuição dos itens em pilhas.



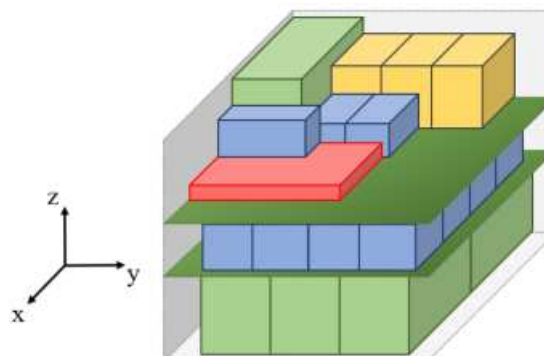
Fonte: o Autor

Ao distribuir os itens em pilhas, podem ocorrer a sobra de espaços acima e ao lado dos itens carregados, que precisam ser verificados para o empacotamento de novos itens e vamos denominar estes espaços de espaços secundários, que são preenchidos seguindo o mesmo procedimento descrito acima.

Empacotamento

O padrão de empacotamento é definido pelo arranjo das camadas completas e incompletas nos compartimentos. O processo de carregamento ocorre pela distribuição das camadas nos compartimentos, empilhadas uma acima da outra e separadas por placas divisórias, como pode ser observado na Figura 4.4. Os compartimentos são acessados diretamente pela lateral do veículo.

Figura 4.4: Exemplo de padrão de empacotamento em um compartimento.

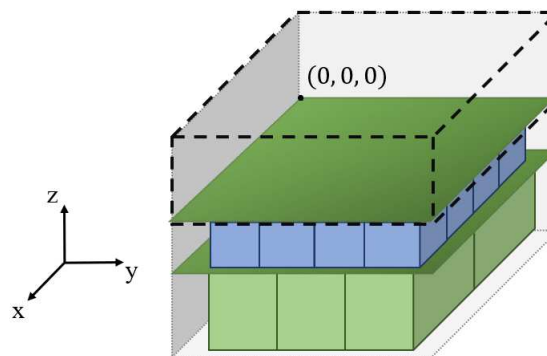


Fonte: Adaptado de Ranck Jr. (2014).

O comprimento e a largura das camadas completas, camadas incompletas e das placas divisórias são consideradas iguais às dimensões dos compartimentos. A altura de uma camada completa é definida pela altura do tipo de item $i \in M$ que compõe esta camada e a altura de uma camada incompleta é dado pela posição mais alta ocupada por um item $i \in M'$ nesta estrutura.

Como as camadas incompletas podem acomodar diferentes tipos de itens $i \in M'$, pode acontecer da camada não possuir uma superfície uniforme (plana), comprometendo a estabilidade dos itens acomodados acima desta estrutura. Para contornar isso, é necessário garantir que as camadas incompletas ocupem sempre a posição mais alta nos compartimentos, como ilustrado na Figura 4.5. Logo, é definido um limitante μ_c para as posições que as camadas completas podem ocupar no compartimento $c \in C$. Este parâmetro é definido em relação a altura do compartimento c , visando garantir o espaço necessário para acomodar a camada incompleta.

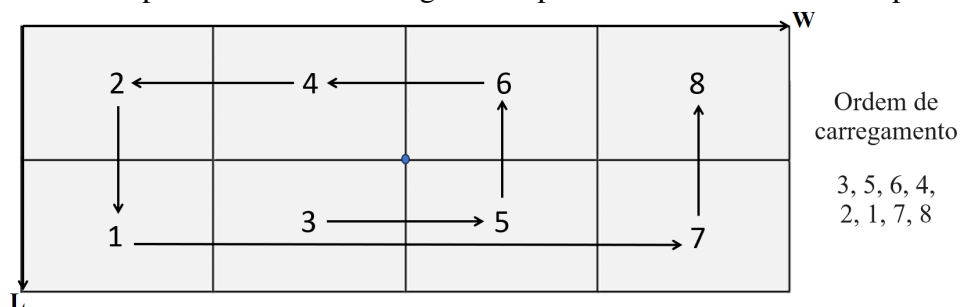
Figura 4.5: Exemplo de espaço destinado a uma camada incompleta.



Fonte: o Autor

A escolha dos compartimentos a serem carregados segue uma ordem pré-definida, denominada ($list^{comp}$), visando favorecer a distribuição balanceada do peso carga no contêiner, adotamos uma sequência espiral conforme apresentado na Figura 4.6. Esta sequência busca carregar primeiramente os compartimentos mais próximos do centro de gravidade do contêiner.

Figura 4.6: Exemplo de ordem de carregamento para um veículo com 8 compartimentos.



Fonte: o Autor.

Os índices e parâmetros considerados na heurística e utilizados nos pseudo-códigos são:

- K : conjunto de destinos das entregas;
- C : conjunto de compartimentos de um contêiner;
- L, W, H : comprimento, largura e altura, respectivamente, do contêiner;
- $L^{comp}, W^{comp}, H_c^{comp}$: comprimento, largura e altura, respectivamente, de um compartimento $c \in C$;
- $list^{comp}$: lista da ordem de carregamento dos compartimentos;
- M : conjunto de itens demandados;
- M' : conjunto de itens residuais;
- N' : conjunto de camadas completas;
- N'' : conjunto de camadas incompletas;
- d_{ik}, d_{ik}^{aux} : demanda do item de tipo i no destino k , com $i \in M, k \in K$;
- Q_i^{max} : quantidade máxima de itens em uma camada completa do tipo $i \in M$;
- Q_{ik}^{mistas} : conjunto com a quantidade de itens do tipo i da parada k alocados em uma camada completa mista, isto é, uma camada completa composta por itens do tipo i de mais de uma parada;
- $Q_{itensPilhas}$: conjunto que armazena a quantidade de itens nas pilhas;
- Q_{pilhas} : conjunto que armazena a quantidade de pilhas nas paredes;
- $pilha^{inc}$: variável de verificação da construção de pilhas incompletas, $pilha^{inc} = 1$ caso tenha gerado pilha incompleta, caso contrário $pilha^{inc} = 0$;
- h_c : ponto mais alto (relativo ao eixo z), ocupado pelas camadas completas acomodadas no compartimento $c \in C$;
- μ_c : limitante superior a maior altura que pode ser ocupada pelas camadas completas no compartimento $c \in C$;
- ψ_i : quantidade mínima de itens em uma camada completa do tipo $i \in M$;
- esp^{livre} : dimensões do espaço livre para acomodar itens em uma camada incompleta, em que $esp^{livre} = (esp^{livre}(x), esp^{livre}(y), esp^{livre}(z))$;

- esp^{altura} : dimensões do espaço secundário altura, localizado acima de uma parede, em que $esp^{altura} = (esp^{altura}(x), esp^{altura}(y), esp^{altura}(z))$;
- $esp^{alturaInc}$: dimensões do espaço secundário altura incompleta de uma parede, em que $esp^{alturaInc} = (esp^{alturaInc}(x), esp^{alturaInc}(y), esp^{alturaInc}(z))$;
- $esp^{lateral}$: dimensões do espaço secundário lateral de uma parede, localizado à direita, em que $esp^{lateral} = (esp^{lateral}(x), esp^{lateral}(y), esp^{lateral}(z))$;
- $esp^{frontal}$: dimensões do espaço secundário frontal de uma parede, em que $esp^{frontal} = (esp^{frontal}(x), esp^{frontal}(y), esp^{frontal}(z))$;
- $nivel$: variável de verificação de nível das paredes, em que $nivel = 0$, o empacotamento de itens ocorre na base da camada incompleta, caso contrário $nivel = 1$;
- $tamanho^{bloco}$: dimensões do espaço ocupado pelos itens $i \in M'$ acomodados na parte completa de um bloco, em que $tamanho^{bloco} = (tamanho^{bloco}(x), tamanho^{bloco}(y), tamanho^{bloco}(z))$;
- $tamanho^{blocoInc}$: dimensões do espaço ocupado pelos itens $i \in M'$ acomodados na parte incompleta de um bloco, em que $tamanho^{blocoInc} = (tamanho^{blocoInc}(x), tamanho^{blocoInc}(y), tamanho^{blocoInc}(z))$;
- $list^{camadas}$: lista de camadas completas que preenchem totalmente um compartimento;
- $list^{verificados}$: lista dos tipos de itens que já foram testados no empacotamento;
- $list^{paredes}$: lista de paredes que compõem uma camada incompleta;
- $list_c^{carga}$: lista de empacotamento dos itens $i \in M'$ no compartimento $c \in C$;

A seguir é detalhada a heurística, descrevendo o processo de construção e carregamento das camadas completas e camadas incompletas, com o auxílio de pseudocódigos.

4.1 CONSTRUÇÃO DAS CAMADAS COMPLETAS

Inicialmente, para cada tipo de item $i \in M$, calcula-se a quantidade máxima de caixas que uma camada completa comporta. Segundo Ranck Jr. (2014), as camadas completas são estruturas com um padrão de empacotamento pré-definido, logo, este processo é considerado como uma etapa de pré-processamento a ser realizada em um momento anterior ao empacotamento dos itens.

A determinação da quantia máxima de itens em uma camada completa pode ser visto como um problema de carregamento bidimensional que deve ser resolvido por um método exato. Para isso, será utilizado o modelo proposto por Beasley (1985) para o problema

de empacotamento de caixas em um palete. Uma revisão da literatura sobre métodos de solução e modelos matemáticos para este tipo de problema pode ser encontrada em Silva, Oliveira e Wäscher (2016).

Para a definição do problema, o modelo associado segue a notação apresentada por Silva, Oliveira e Wäscher (2016), considerando o objeto como sendo um palete, de dimensões (L, W) , e um conjunto de itens, de dimensões (a, b) , a serem acomodados sobre o palete, sem que caixas sejam empilhadas, com $L, W, a, b \in \mathbb{Z}$. Os itens podem ser empacotados de acordo com duas orientações: a orientação original ($i = 1$) e com rotação de 90° ($i = 2$). As coordenadas (p, q) determinam as possíveis posições a serem ocupadas pelos itens no palete e definidas pelo conjunto:

$$P = \{(p, q) : p \in X, q \in Y\}, \quad X = \{0, 1, 2, \dots, L - 1\} \text{ e } Y = \{0, 1, 2, \dots, W - 1\}. \quad (4.1)$$

Podemos afirmar que um item de dimensões (a_i, b_i) , com orientação $i = 1, 2$, está acomodado no palete quando o canto inferior esquerdo do item é posicionado no ponto (p, q) no sistema de coordenadas estabelecido para o palete. O canto inferior esquerdo é o ponto de referência do item. Para essa avaliação, é definida a variável de decisão x_{ipq} , que é descrita da seguinte maneira:

$$x_{ipq} = \begin{cases} 1, & \text{se o canto inferior esquerdo do item com orientação } i \text{ (} i = 1, 2 \text{) é alocado} \\ & \text{na posição } (p, q) \text{ sendo } p \in X \wedge p \leq L - a_i \text{ e } q \in Y \wedge q \leq W - b_i \\ 0, & \text{caso contrário.} \end{cases} \quad (4.2)$$

No contexto do empacotamento, a fim de evitar a sobreposição de itens, é essencial que, quando o canto inferior do item é posicionado em (p, q) , as posições internas (r, s) do item estejam indisponíveis para acomodar qualquer outro item. Para alcançar esse objetivo, é estabelecido o parâmetro u_{ipqrs} , que é definido da seguinte forma:

$$u_{ipqrs} = \begin{cases} 1, & \text{se } 0 \leq p \leq r \leq p + a_i - 1 \leq L - 1 \text{ e } 0 \leq q \leq s \leq q + b_i - 1 \leq W - 1, \\ & p, r \in X \wedge p, r \leq L - a_i, q, s \in Y \wedge q, s \leq W - b_i \text{ e } i = 1, 2 \\ 0, & \text{caso contrário.} \end{cases} \quad (4.3)$$

O modelo de Beasley (1985) para este problema é definido por:

$$\text{Max} \sum_{i=1}^2 \sum_{\substack{p \in X \wedge \\ p \leq L - a_i}} \sum_{\substack{q \in Y \wedge \\ q \leq W - b_i}} x_{ipq} \quad (4.4)$$

$$\sum_{i=1}^2 \sum_{\substack{p \in X \wedge \\ p \leq L - a_i}} \sum_{\substack{q \in Y \wedge \\ q \leq W - b_i}} u_{ipqrs} x_{ipq} \leq 1, \quad \forall r \in X, s \in Y \quad (4.5)$$

$$x_{ipq} \in \{0, 1\}, \quad \begin{matrix} \forall i=1,2 \\ p \in X \wedge p \leq L - a_i \\ q \in Y \wedge q \leq W - b_i. \end{matrix} \quad (4.6)$$

A função objetivo (4.4) maximiza a quantidade de caixas empacotadas no palete. As restrições (4.5) garantem a não sobreposição de itens no palete e as restrições (4.6) determinam o domínio das variáveis de decisão.

Observando o modelo (4.4) - (4.6), o número de variáveis pode ser muito grande, visto que, as variáveis definidas em (4.6) são observadas para cada ponto do sistema de coordenadas do palete. Para contornar isso, é possível considerar a discretização proposta por Herz (1972) e Christofides e Whitlock (1977), em que os conjuntos X e Y definidos em (4.1) são substituídos, respectivamente, pelos conjuntos normais:

$$X' = \{l : l = \alpha \cdot a_i + \beta \cdot b_i, 0 \leq l \leq L - a, \alpha, \beta \in \mathbb{Z}_+ \cup \{0\}\} \quad (4.7)$$

$$Y' = \{w : w = \alpha \cdot a_i + \beta \cdot b_i, 0 \leq w \leq W - b, \alpha, \beta \in \mathbb{Z}_+ \cup \{0\}\} \quad (4.8)$$

Os conjuntos definidos em (4.7) e (4.8) consideram as posições disponíveis no palete com suas coordenadas definidas como uma combinação inteira não negativa do comprimento e largura do item a ser acomodado. Dessa forma, a quantidade de pontos (p, q) é reduzida, diminuindo assim a quantidade de variáveis do modelo.

Tendo determinada as quantidades máximas de itens em uma camada completa para cada tipo de item demandado, que podem ser observadas no Anexo A deste trabalho, a construção destas estruturas segue a demanda d_{ik} de cada item do tipo $i \in M$ para cada destino $k \in K$ da lista de paradas. Inicialmente, as camadas são montadas utilizando a quantidade demandada por apenas um cliente $k \in K$, em que tenta-se gerar o máximo de camadas com a quantidade de itens na parada analisada. Após este primeiro processo, uma nova tentativa de formar camadas completas é realizada, agora unindo as demandas de diversas paradas $k \in K$, respeitando a quantidade máxima das camadas completas de cada tipo $i \in M$. Para que um conjunto de itens forme uma nova camada, é necessário que a quantidade de itens da última parada garantam a estabilidade da carga acomodada acima deles. Esta condição define que os últimos itens a serem entregues, tenham uma quantidade maior ou igual ao número mínimo de itens (ψ_i) para uma camada completa, a fim de assegurar o suporte necessário para os itens alocados acima. Para identificação, definimos as camadas geradas com demanda de uma única

parada como sendo do tipo simples, e camadas geradas com demanda de mais de uma parada como sendo do tipo mista.

Com a etapa de construção das camadas completas (N') finalizada, o carregamento decorre por empacotar as camadas seguindo a ordem inversa da lista de paradas, buscando que os itens das primeiras camadas a serem carregadas sejam os últimos a serem entregues. A ordem de carregamento dos compartimentos segue uma lista pré-definida em $list^{comp}$. Para garantir um melhor empacotamento das camadas, uma boa distribuição do peso da carga e carregamento de todos os itens demandados, foi considerado a utilização total do espaço dos compartimentos centrais do veículo ($c = 3, 4, 5$ e 6)⁴. Para os demais compartimentos ($c = 1, 2, 7$ e 8) foi considerado um limitante μ_c para o espaço disponível para o carregamento das camadas completas. Dessa forma, é garantido um espaço destinado ao empacotamento das camadas incompletas.

Determinar um conjunto de camadas completas que preencham totalmente um compartimento $c \in C$, é um processo que pode ser encarado como um subproblema, a ser resolvido como um problema da mochila unidimensional. Este problema envolve a escolha de um conjunto de m itens, cada um com um valor v_i , um peso associado l_i e uma demanda d_i , de tal forma que a soma dos valores dos itens empacotados seja maximizada, respeitando uma capacidade máxima do peso considerado da mochila (L). A formulação matemática é dada por:

$$Max \quad \sum_{i=1}^m v_i a_i \quad (4.9)$$

$$\sum_{i=1}^m l_i a_i \leq L \quad (4.10)$$

$$0 \leq a_i \leq d_i, \text{ inteiro}, \quad i = 1, \dots, m \quad (4.11)$$

O problema da mochila é resolvido para cada um dos compartimento $c = 3, 5, 6, 4$, definindo assim a lista de camadas completas ($list^{camadas}$) a serem empacotadas no compartimento c . Entretanto, foi definido $l_i = v_i$ e o problema é conhecido como *subset sum problem*. O carregamentos dos demais compartimentos ocorre com a tentativa de alocar uma camada completa $j \in N'$ em cada compartimento da sequência, respeitando as restrições de resistência ao empilhamento. Após o último compartimento ser verificado, a sequência reinicia e o carregamento das camadas continua, percorrendo novamente a lista. Neste processo, se a altura das camadas completas dentro de um compartimento c atingir o limitante μ_c , este compartimento é marcado como fechado e nenhuma outra camada completa é atribuída a ele.

Além disso, caso uma camada completa $j \in N'$ viole a restrição de resistência ao empilhamento, mostrando que as camadas acomodadas abaixo dela não sustentam a pressão exercida por esta camada j , ela então não é carregada no compartimento $c \in C$, sendo mantida em N' . O empacotamento do compartimento c segue com a próxima camada $j' > j \in N'$.

⁴A identificação dos compartimentos segue a numeração apresentada na Figura 4.6.

O processo de empacotamento continua até que não hajam mais camadas no conjunto N' a serem empacotadas ou o limitante μ_c , para o espaço destinado as camadas completas, de todos os compartimentos tenham sido atingidos. Havendo alguma camada completa $j \in N'$ que não foi carregada em nenhum dos compartimentos, os itens desta camada são repassados para o conjunto de itens residuais M' .

O Algoritmo 1 apresenta o procedimento responsável por gerar as camadas completas, verificando a partir da demanda d_{ik} , do item de tipo $i \in M$ com a quantidade solicitada pelo cliente da parada $k \in K$, quantas camadas completas simples podem ser geradas, correspondendo às linhas 9 a 17. A segunda etapa do algoritmo é responsável por gerar as camadas completas mistas, unindo demandas de diversas paradas $k \in K$, presente nas linhas 18 a 40.

O Algoritmo 2 apresenta o procedimento encarregado do carregamento das camadas completas nos compartimentos do veículo, seguindo a ordem inversa das paradas. Os compartimentos são diferenciados entre os que serão carregados ao máximo e aqueles que adotam um limitante para o espaço disponível das camadas completas. Para os compartimentos centrais ($c = 3, 4, 5, 6$), é resolvido o problema da mochila (linhas 9 e 10), a fim de determinar um conjunto de camadas completas que preencherão o compartimento c , correspondendo às linhas 13 a 25. Para os demais compartimentos ($c = 1, 2, 7, 8$), o carregamento das camadas ocorre de modo a acomodar o maior número possível de camadas em cada compartimento, $c \in C$, respeitando as restrições de resistência ao empilhamento, correspondendo às linhas 26 a 39.

Algoritmo 1: Procedimento para gerar camadas completas

```

1  início
2  Dados de Entrada:  $M, K, Q_i^{max}, d_{ik}$ ;
3  Dados de Saída:  $N', list_c^{carga}$ ;
4  Definir o conjunto de camadas completas  $N' \leftarrow \emptyset$ ;
5  Definir a lista de demandas auxiliar  $d_{ik}^{aux} \leftarrow d_{ik}$ ;
6  Definir a lista da quantidade de itens na camada completa mista  $Q_{ik}^{mistas} \leftarrow \emptyset$ ;
7  // Conjunto com a quantidade de itens  $i$  da parada  $k$  em
   uma camada completa mista
8  Definir a variável de controle da quantidade de itens em uma camada completa
   mista  $list_{qtd} \leftarrow \emptyset$ ;
9  para ( $k = |K|, \dots, 1$ ) faça
10     para ( $i \in M$ ) faça
11         Calcular quantas camadas completas podem ser geradas com a demanda
            $d_{ik}^{aux}$ ;
12         Salvar no conjunto  $N'$ , as camadas completas geradas com  $d_{ik}^{aux}$ ;
13         Salvar o peso total da camada;
14         Salvar resistência da camada ao empilhamento de itens;
15         Atualizar lista de demandas  $d_{ik}^{aux}$ , descontando a quantidade de itens  $i$  da
           camada completa;
16     fim
17 fim
18 para ( $i \in M$ ) faça
19      $list_{qtd} \leftarrow \emptyset$ ;
20     // Zerar variável de controle da quantidade de itens
     na camada completa mista a ser criada
21     para ( $k = |K|, \dots, 1$ ) faça
22         se ( $d_{ik}^{aux} \geq \psi_i$ ) então
23             Salvar em  $Q_{ik}^{mistas}$  a quantidade de itens do tipo  $i$  da parada  $k$ 
               atribuídos a camada completa mista;
24             Salvar em  $list_{qtd}$  a quantidade de itens em  $d_{ik}^{aux}$ , alocados na camada
               completa mista;
25             Atualizar lista de demandas  $d_{ik}^{aux}$ , descontando a quantidade de itens
               acomodados na camada completa mista;
26             para ( $k' = k - 1, \dots, 1$ ) faça
27                 se ( $d_{ik'}^{aux} > 0$ ) então
28                     Calcular a quantidade de itens a ser utilizada de  $d_{ik'}^{aux}$  na
                       montagem da camada completa mista;
29                     Salvar em  $Q_{ik'}^{mistas}$  a quantidade de itens do tipo  $i$  da parada
                        $k'$  atribuídos a camada completa mista;
30                     se ( $list_{qtd} = Q_i^{max}$ ) então
31                         Salvar a camada completa gerada no conjunto  $N'$ ;
32                         Salvar o peso total da camada;
33                         Salvar resistência da camada ao empilhamento de itens;
34                         Atualizar lista de demandas  $d_{ik'}^{aux}$ , descontando a
                           quantidade de itens acomodados na camada completa
                           mista;
35                     fim
36                 fim
37             fim
38         fim
39     fim
40 fim
41 Ordenar conjunto  $N'$ , de camadas completas, em ordem decrescente de altura;
42 fim

```

Algoritmo 2: Procedimento para carregar as camadas completas.

```

1  início
2  Dados de Entrada:  $K, C, N', \mu_c, list^{comp}$ ;
3  Dados de Saída:  $list_c^{carga}, M'$ ;
4  Definir o conjunto de itens residuais  $M' \leftarrow \emptyset$ ;
5  Definir  $h_c = 0$ , para todo  $c \in C$ ; // Altura da carga em  $c \in C$ 
6  Definir a lista de itens carregados no compartimento  $c$  ( $list_c^{carga}$ );
7  // Lista com os itens carregados no compartimento  $c \in C$ 
8  para ( $c \in list^{comp}$ ) faça
9      Resolver o problema da mochila para o compartimento  $c$ ;
10     Salvar em  $list^{camadas}$  a sequência de camadas completas a serem carregadas
        no compartimento;
11     para ( $k = |K|, \dots, 1$ ) faça
12         para ( $j \in N'$ ) faça
13             se (O compartimento  $c$  é um dos compartimentos centrais) então
14                 se (Existe uma camada completa do tipo  $j \in N'$  com itens da
                    parada  $k$ ) então
15                     se (A camada completa do tipo  $j$  está em  $list^{camadas}$ ) então
16                         Verificar se o volume de carga transportado pelo
                            contêiner não é extrapolado;
17                         Verificar se a resistência ao empilhamento das camadas
                            empacotadas anteriormente é satisfeita;
18                         se (As verificações foram satisfeitas) então
19                             Salvar dados da camada carregada em  $list_c^{carga}$ ;
20                             Atualizar conjunto das camadas completas  $N'$ 
21                             Atualizar a demanda  $d_{ik}$ , descontando a quantidade
                                    de itens  $j$  na camada completa;
22                             fim
23                         fim
24                     fim
25                 fim
26             se (O compartimento  $c$  não é um dos compartimentos centrais) então
27                 se (Existe uma camada completa do tipo  $j \in N'$  com itens da
                    parada  $k$ ) então
28                     Verificar se  $h_c + h_j \leq \mu_c$ ;
29                     // A camada completa cabe no em  $c$ 
30                     Verificar se o volume de carga transportado pelo contêiner
                            não é extrapolado;
31                     Verificar se a resistência ao empilhamento das camadas
                            empacotadas anteriormente é satisfeita;
32                     se (As verificações foram satisfeitas) então
33                         Salvar dados da camada carregada em  $list_c^{carga}$ ;
34                         Atualizar a altura disponível no compartimento:
                                 $h_c = h_c + h_j$ ;
35                         Atualizar conjunto das camadas completas  $N'$ 
36                         Atualizar a demanda  $d_{ik}$ , descontando a quantidade de
                                itens  $j$  na camada completa;
37                         fim
38                     fim
39                 fim
40             fim
41         fim
42     fim
43     Adicionar os itens das camadas completas  $j \in N'$  não carregadas no conjunto de
        itens residuais  $M'$ ;
44     Ordenar o conjunto  $M'$ , em ordem decrescente pela altura dos itens;
45 fim

```

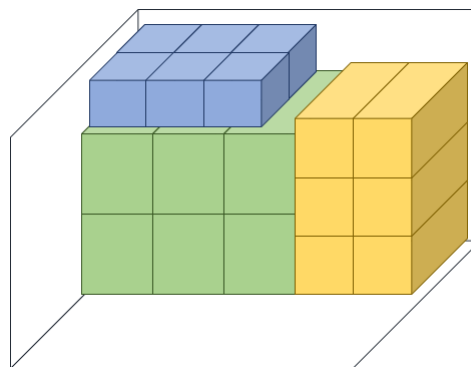
4.2 CONSTRUÇÃO DAS CAMADAS INCOMPLETAS

Com o conjunto de itens residuais definido após o carregamento das camadas completas, se inicia a construção da camada incompleta para os compartimentos $c \in C$ em que foram considerados o limitante de espaço μ_c . O processo de construção das camadas incompletas é baseado no procedimento de carregamento apresentado por Moura e Oliveira (2005), em que o empacotamento das caixas ocorre por acomodar os itens residuais $i \in M'$ em paredes verticais. O carregamento ocorre até que o contêiner não tenha mais espaço livre disponível, não existam mais caixas a serem carregadas ou as dimensões do espaço livre sejam menor que as dimensões do menores item. É importante destacar que, para que a solução obtida seja factível, é necessário que todos os itens demandados sejam carregados.

O conjunto de itens residuais M' contém os itens que não foram acomodados nas camadas completas e as demandas de cada parada $k \in K$ são armazenadas em d_{ik} . A seleção das caixas segue a lista de paradas em ordem inversa, para que novamente os primeiros itens a serem entregues sejam os últimos a serem carregados. Inicialmente, os itens residuais são ordenados pela altura das caixas em ordem decrescente, e caso ocorra de itens terem a mesma altura, como critério de desempate, é considerado o peso das caixas, em ordem decrescente. A estratégia gulosa para seleção escolhe sempre o primeiro item da lista, permitindo assim que os maiores itens tenham mais possibilidades de serem carregados.

Os itens são dispostos nas camadas incompletas em paredes, que são montadas alocando os itens em blocos, conforme pode ser observado na Figura 4.7. Uma parede pode ser composta por blocos de diferentes tipos de itens $i \in M'$, sendo um bloco composto com a demanda de uma única parada $k \in K$.

Figura 4.7: Exemplo de parede.

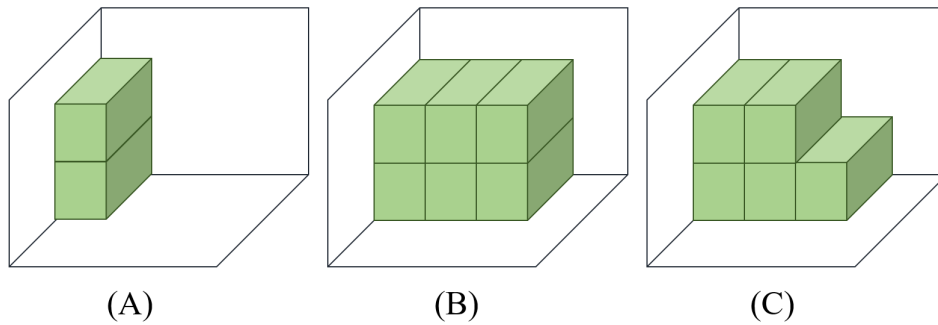


Fonte: o Autor

A construção de um bloco ocorre por acomodar os itens em uma pilha, com a maior quantidade de itens possível, como apresentado na Figura 4.8 (A), respeitando as dimensões do compartimento c . Em seguida, esta pilha é replicada na largura do compartimento, até preencher o espaço disponível para construção da parede, como ilustrado na Figura 4.8 (B).

O comprimento de uma pilha é dado pelo comprimento do item i que a com-

Figura 4.8: Exemplos de pilha e blocos.

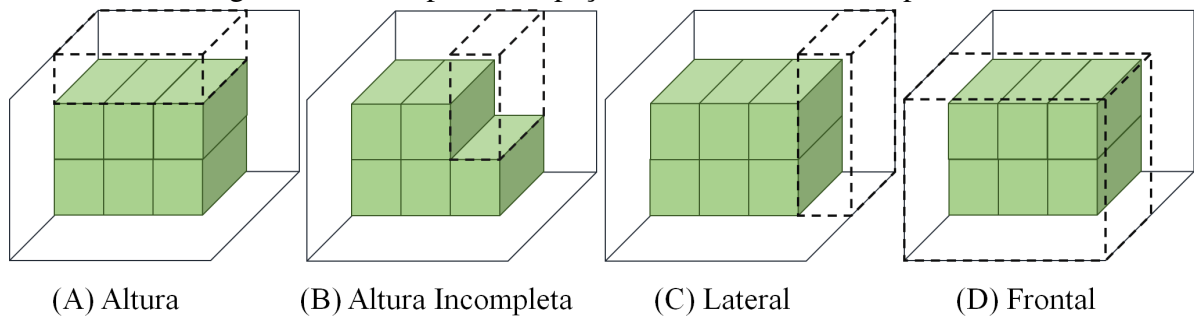


Fonte: o Autor

põe, a largura é dada pelo produto da largura do item i pela quantidade de pilhas (Q^{Pilhas}) e a altura é dada pela altura do item i multiplicada pela quantidade de itens em uma pilha ($Q^{ItensPilha}$). Ademais, pode ocorrer de uma parada $k \in K$ não ter demanda suficiente para montar uma parede completamente, fazendo com que a última pilha da parede tenha menos itens que as demais, com isso, a parede é tida como incompleta, como exemplificado na Figura 4.8 (C). O tamanho do bloco de itens é diferenciado pela parte completa ($tamanho^{bloco}$) e pela parte incompleta ($tamanho^{blocoInc}$).

Com o carregamento de um bloco, espaços secundários são gerados na parede e precisam ser verificados no empacotamento de novos itens. Estes espaços considerados são observados acima do bloco (esp^{altura}), acima da parte incompleta do bloco ($esp^{alturaInc}$), a direita do bloco ($esp^{lateral}$) e a frente do bloco ($esp^{frontal}$), como ilustrados na Figura 4.9.

Figura 4.9: Exemplo dos espaços secundários em uma parede.

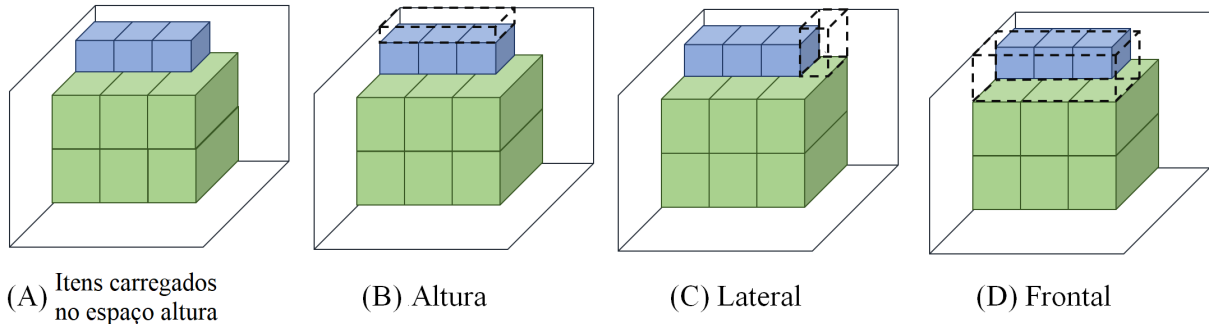


Fonte: o Autor

Os espaços secundários são verificados na ordem: altura, altura incompleta, lateral e frontal; e o empacotamento de itens $i \in M'$ nestes espaços são realizados repetindo o procedimento de construção de blocos. Conforme os itens são distribuídos a estes espaços, uma nova divisão é feita e novos espaços são gerados, definidos como um nível dois de espaços secundários. Na Figura 4.10 é possível observar um exemplo de espaços secundários em um segundo nível a partir de itens $i \in M'$ acomodados em um espaço altura.

Este segundo nível é verificado em sua totalidade, tentando acomodar itens $i \in M'$ em todas as regiões secundárias geradas. Ao fim deste ciclo, continua-se o empaco-

Figura 4.10: Exemplo de um segundo nível de espaços secundários em uma parede.



Fonte: o Autor

tamento para o nível anterior de espaços secundários com a próxima região, até que todos os espaços sejam verificados. Este ciclo continua até que não tenham mais itens residuais $i \in M'$ a serem empacotados, ou não seja possível alocar itens no espaço livre destinado para a camada incompleta do compartimento $c \in C$, então o próximo compartimento da $list^{comp}$ é selecionado e se inicia a construção de uma nova camada incompleta.

O Algoritmo 3 exibe o procedimento responsável por comandar a construção da camada incompleta em cada compartimento $c \in C$, correspondendo às linhas 6 a 14. Ademais, o procedimento verifica se houve espaço não utilizado na região destinada ao carregamento das camadas completas e armazena o espaço total disponível para construção da camada incompleta em esp^{livre} , linha 4. Para cada compartimento c , é definida uma lista de paredes ($list^{paredes}$) que formam uma camada incompleta, presente na linha 5.

Algoritmo 3: Procedimento para criação das camadas incompletas em cada compartimento.

```

1 início
2   Dados de Entrada:  $M', d_{ik}, list_c^{carga}, h_c, list^{comp}$ ;
3   para  $c \in list^{comp}$  faça
4     Armazenar o espaço livre disponível para montar a camada incompleta no
       compartimento  $c \in C$  na variável  $esp^{livre}$ ;
5     Definir lista de paredes que compõem a camada incompleta  $list^{paredes} \leftarrow \emptyset$ ;
6     se (Existe espaço livre no compartimento c) então
7        $nivel = 0$ ; // Indica que espaço livre inicial está
          na base da camada incompleta
8       se ( $M' \neq \emptyset$ ) então
9         // Há itens residuais a serem empacotados
10        Executar Algoritmo 4 ( $M', d_{ik}, c \in C, esp^{livre}, nivel, list^{paredes}$ );
11        //Procedimento que controla o espaço
          disponível no compartimento  $c \in C$ , para alocar
          itens na camada incompleta;
12      fim
13    fim
14    Salvar em  $list_c^{carga}$  as paredes geradas ( $list^{paredes}$ ) para a camada incompleta;
15  fim
16 fim

```

O Algoritmo 4 apresenta o procedimento encarregado de controlar o preenchimento de uma camada incompleta no compartimento $c \in C$ enquanto houver o espaço disponível para construção de paredes. Para isso, são definidas inicialmente as menores dimensões observadas entre os itens $i \in M'$ a serem consideradas como o espaço mínimo necessário para acomodar um item, presente nas linhas 3 e 4. A construção de uma camada incompleta continua enquanto o comprimento (eixo x) do espaço disponível couber algum item $i \in M'$, correspondendo às linhas 5 a 10. A seleção dos itens a serem empacotados segue as demandas na ordem inversa de paradas, considerando os itens ordenados pela altura de maneira decrescente.

Algoritmo 4: Procedimento de controle do espaço livre no compartimento c para construção de paredes. ($M', d_{ik}, c \in C, esp^{livre}, nivel, list^{paredes}$)

```

1 início
2   Dados de entrada:  $M', d_{ik}, c \in C, esp^{livre}, nivel, list^{paredes},$ 
3    $dim^{min} = \min\{w_i, l_i, i \in M'\}$  e  $alt^{min} = \min\{h_i, i \in M'\};$ 
4   enquanto (Existir um item  $i \in M'$  que caiba no  $esp^{livre}$ ) faça
5     Executar Algoritmo 5 ( $M', d_{ik}, c \in C, esp^{livre}, nivel, list^{paredes}$ );
6     // Procedimento de controle do empacotamento de
7     itens no  $esp^{livre}$ 
8     Incrementar contagem de paredes;
9     Atualizar o  $esp^{livre}$  disponível no compartimento  $c \in C$  para montagem da
10    camada incompleta;
11  fim
12 fim

```

O Algoritmo 5 exibe o procedimento responsável pelo preenchimento do espaço livre no compartimento $c \in C$, para construção de uma parede. Inicialmente, o procedimento recebe como parâmetro do Algoritmo 4 o espaço livre inicial para a camada incompleta (esp^{livre}). Ao alocar itens nesta região, o Algoritmo 6 é executado para o cálculo dos espaços secundários, presente na linha 9. O Algoritmo 7 calcula a formação dos espaços secundários, presente na linha 12. Para cada região secundária o Algoritmo 5 é executado novamente, de maneira recursiva, passando como parâmetro o espaço livre associado a região analisada e tentando o carregamento dos itens, verificando o espaço altura nas linhas 14 a 17, o espaço altura incompleta nas linhas 18 a 21, o espaço lateral nas linhas 22 a 24 e o espaço frontal nas linhas 25 a 27. É possível que um item $i \in M'$ caiba em todos os espaços secundários gerados, o que determina este carregamento é a existência de demanda deste item, verificada na etapa seguinte.

Algoritmo 5: Procedimento de controle do espaço livre para construção de blocos ($M', d_{ik}, c \in C, esp^{livre}, nivel, list^{paredes}$).

```

1 início
2   Dados de Entrada:  $M', d_{ik}, c \in C, esp^{livre}, nivel, list^{paredes}$ ;
3   Definir as variáveis  $esp^{Altura}, esp^{AlturaInc}, esp^{Lateral}, esp^{Frontal}$ ;
4   // Variáveis para controle dos espaços secundários
5   enquanto (Existir um item  $i \in M'$  que cabe em  $esp^{livre}$ ) faça
6     Definir a variável  $tamanho^{bloco} \leftarrow \emptyset$ ;
7     Definir a variável  $tamanho^{blocoInc} \leftarrow \emptyset$ ;
8     // Variáveis de controle do tamanho do bloco de
9     // itens formado
10    Executar Algoritmo 6 ( $M', d_{ik}, esp^{livre}, nivel, tamanho^{bloco},$ 
11     $tamanho^{blocoInc}, list^{paredes}$ );
12    // Procedimento para construir um bloco de itens
13    se (Itens foram alocados na parede) então
14      Executar Algoritmo 7 ( $esp^{livre}, nivel, tamanho^{bloco}, tamanho^{blocoInc},$ 
15       $esp^{Altura}, esp^{AlturaInc}, esp^{Lateral}, esp^{Frontal}$ );
16      // Procedimento para calcular os espaços
17      // secundários gerados com os itens alocados
18      se (Existir um item  $i \in M'$  que cabe em  $esp^{altura}$ ) então
19         $nivel = 1$ ; // Indicar que espaço altura não está
20        // na base da camada incompleta
21        Executar Algoritmo 5 ( $M', d_{ik}, c \in C, esp^{altura}, nivel,$ 
22         $list^{paredes}$ ); // Procedimento de controle do
23        // empacotamento de itens no  $esp^{altura}$ 
24      fim
25      senão se (Existir um item  $i \in M'$  que cabe em  $esp^{alturaInc}$ ) então
26         $nivel = 1$ ; // Indicar que espaço altura
27        // incompleta não está na base da camada
28        // incompleta
29        Executar Algoritmo 5 ( $M', d_{ik}, c \in C, esp^{alturaInc}, nivel,$ 
30         $list^{paredes}$ ); // Procedimento de controle do
31        // empacotamento de itens no  $esp^{alturaInc}$ 
32      fim
33      senão se (Existir um item  $i \in M'$  que cabe em  $esp^{lateral}$ ) então
34        Executar Algoritmo 5 ( $M', d_{ik}, c \in C, esp^{lateral}, nivel,$ 
35         $list^{paredes}$ ); // Procedimento de controle do
36        // empacotamento de itens no  $esp^{lateral}$ 
37      fim
38      senão se (Existir um item  $i \in M'$  que cabe em  $esp^{frontal}$ ) então
39        Executar Algoritmo 5 ( $M', d_{ik}, c \in C, esp^{frontal}, nivel,$ 
40         $list^{paredes}$ ); // Procedimento de controle do
41        // empacotamento de itens no  $esp^{frontal}$ 
42      fim
43    fim
44  fim
45 fim

```

O Algoritmo 6 apresenta o procedimento responsável pelo empacotamento dos itens $i \in M'$ no espaço livre selecionado. O procedimento recebe como parâmetro o espaço a ser considerado (esp^{livre}) e testa o empacotamento dos itens i demandados em cada parada $k \in K$. A seleção de uma caixa i segue a lista de demandas na ordem inversa e seleciona o primeiro item do conjunto de residuais (M'), que estão ordenados de maneira decrescente pela altura. Como critério de desempate, é considerado o peso dos itens em ordem decrescente. No carregamento, é verificada a orientação que permite acomodar o maior número possível de itens na largura (y) no espaço livre. Após todos os itens terem sido verificados com suas melhores orientações, a tentativa de empacotamento recomeça, considerando as segundas orientações dos itens i , até que seja determinado um item que caiba no espaço livre disponível, presente na linha 13.

Com um item determinado, calcula-se quantas caixas cabem na pilha de itens e em seguida, verifica-se quantas pilhas cabem no espaço livre, em relação a largura, correspondendo às linhas 17 a 19. Com o cálculo das pilhas realizado, são armazenadas as dimensões da parte completa ($tamanho^{bloco}$) e incompleta ($tamanho^{blocoInc}$) para verificação dos espaços secundários, presente na linha 20. Com o bloco de itens definido, o procedimento verifica as posições ocupadas por cada item na parede, a partir do sistema de coordenadas definido para a camada incompleta no compartimento $c \in C$, sendo verificado nas linhas 24 a 26. Para cada item alocado, as restrições são verificadas. As restrições de volume do veículo e resistência ao empilhamento não sendo violadas, os dados do item i são armazenados em $list^{paredes}$, correspondendo às linhas 27 a 30.

Algoritmo 6: Procedimento para empacotamento dos itens em pilhas (M' , d_{ik} , esp^{livre} , $nivel$, $tamanho^{bloco}$, $tamanho^{blocoInc}$, $list^{paredes}$)

```

1 início
2   Dados de Entrada:  $M'$ ,  $d_{ik}$ ,  $esp^{livre}$ ,  $nivel$ ,  $tamanho^{bloco}$ ,  $tamanho^{blocoInc}$ ,
    $list^{paredes}$ ;
3   Definir a variável  $pilha^{inc}$ , para controle da construção do bloco de itens;
4   Definir a variável auxiliar  $tamanho^{total} \leftarrow \emptyset$ 
5   // Variável para verificar se o bloco criado cabe no
   espaço livre
6   enquanto (Todos os itens não foram verificados com as duas orientações) faça
7     Definir lista de itens verificados  $list^{verificados} \leftarrow \emptyset$ ;
8     para  $k = |K|, \dots, 1$  faça
9       para  $i \in M'$  faça
10        se ( $d_{ik} > 0$ ) então
11          Marcar tipo de item  $i$  na lista  $list^{verificados}$  como verificado;
12          Verificar a melhor orientação para empacotamento do item  $i$ ;
13          se (O item  $i$  já foi verificado com a melhor orientação) então
14            Considerar a segunda orientação para empacotamento do
            item  $i$ ;
15            // Orientação com rotação de 90° no plano
             $xy$  em relação a melhor orientação;
16          fim
17          Verificar quantos itens cabem na pilha no  $esp^{livre}$ , em relação a
            altura disponível, e armazenar em  $Q^{itensPilha}$ ;
18          Verificar quantas pilhas no  $esp^{livre}$ , em relação a largura
            disponível, e armazenar em  $Q^{Pilhas}$ ;
19          Verificar se foi gerada pilha incompleta,  $pilha^{inc} = 1$ , caso
            contrário  $pilha^{inc} = 0$ ;
20          Armazenar o espaço ocupado pelos itens em  $tamanho^{bloco}$  ou
             $tamanho^{blocoInc}$ ;
21          se (Espaço ocupado pelos itens cabe em  $esp^{livre}$  para as
            dimensões  $x$ ,  $y$  e  $z$ ) então
22            para ( $p = 1, \dots, Q^{Pilhas}$ ) faça
23              para ( $n = 1, \dots, Q^{itensPilha}$ ) faça
24                Verificar coordenadas da posição em que o item  $i$ 
                da posição  $(p, n)$  será acomodado na parede;
25                Verificar volume de carga no contêiner;
26                Verificar restrições de resistência ao empilhamento
                para o item  $i$  na posição  $(p, n)$  da parede;
27                se (As restrições não são violadas) então
28                  Salvar dados do item alocado no bloco em
                   $list^{paredes}$ ;
29                  Atualizar demanda  $d_{ik}$  e conjuntos de itens
                  residuais  $M'$  descontando os itens
                  carregados;
30                fim
31              fim
32            fim
33          fim
34        fim
35      fim
36    fim
37  fim
38 fim

```

O Algoritmo 7 realiza o cálculo dos espaços secundários após o empacotamento de itens. A divisão dos espaços é calculada a partir dos itens acomodados no espaço livre inicial (esp^{livre}) considerado no empacotamento dos itens $i \in M'$ na parede, recebido como parâmetro do Algoritmo 5. O espaço ocupado pelas caixas carregadas $tamanho^{bloco}$ e $tamanho^{blocoInc}$ é passado como parâmetro para o procedimento descrito no Algoritmo 7 para a divisão dos espaço secundários. A primeira parte do algoritmo é responsável por verificar a existência, ou não, de pilhas incompletas no primeiro bloco de itens acomodados na parede, correspondendo às linhas 2 a 13. Na segunda parte, são verificados os espaços secundários dos blocos de itens seguintes, calculando as dimensões dos quatro espaços secundários possíveis, presente nas linhas 14 a 16.

Algoritmo 7: Procedimento para calcular os espaços secundários. (esp^{livre} , $nivel$, $tamanho^{bloco}$, $tamanho^{blocoInc}$, esp^{altura} , $esp^{alturaInc}$, $esp^{lateral}$, $esp^{frontal}$)

```

1 início
2   se ( $nivel = 0$ ) então
3     // Os itens carregados foram acomodados na base da
4     // camada incompleta
5     se ( $pilha^{inc} = 0$ ) então
6       // Não foi gerada uma pilha incompleta nos itens
7       // carregados
8       Calcular espaços livres secundários ( $esp^{altura}$ ,  $esp^{lateral}$ ,  $esp^{frontal}$ ) a
9       // partir de  $tamanho^{bloco}$ ;
10       $esp^{alturaInc} \leftarrow 0$ ; // Espaço secundário altura
11      // incompleta é zero
12     fim
13   fim
14   se ( $nivel > 0$ ) então
15     // Foi gerada uma pilha incompleta nos itens
16     // carregados
17     Calcular espaços livres secundários ( $esp^{altura}$ ,  $esp^{alturaInc}$ ,  $esp^{lateral}$ ,
18     //  $esp^{frontal}$ ) a partir de  $tamanho^{blocoInc}$ ;
19     fim
20   fim
21 fim

```

O espaço altura (esp^{altura}) é considerado pela região localizada acima dos itens alocados na parte completa da parede, como apresentado na Figura 4.9 (A), dado pela diferença da altura do espaço livre (esp^{livre}) e a altura ocupado pelos itens em $tamanho^{parede}$.

O espaço altura incompleta ($esp^{alturaInc}$) é considerado pela região situada acima dos itens na parte incompleta da parede, como pode ser observado na Figura 4.9 (B), dado pela diferença da altura do espaço livre (esp^{livre}) e a altura ocupado pelos itens em $tamanho^{parede}$.

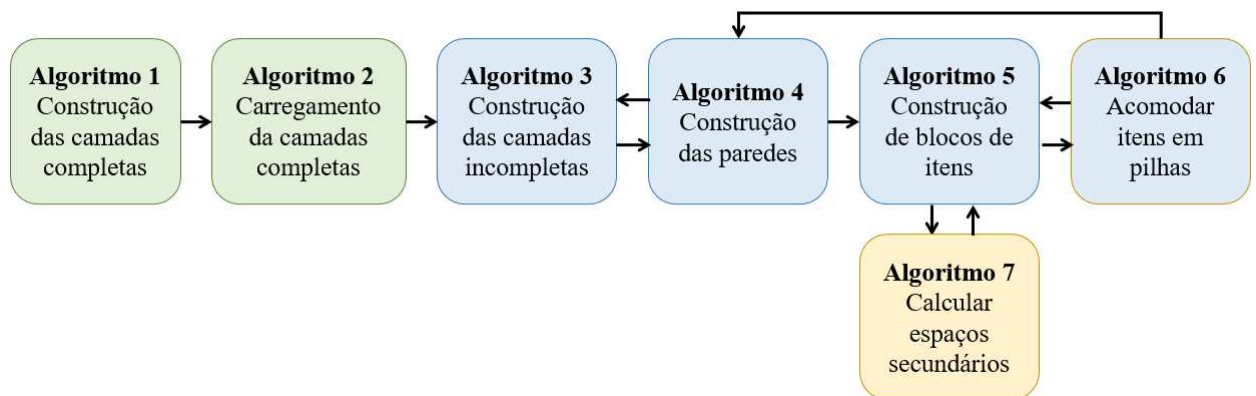
O espaço lateral ($esp^{lateral}$) é considerado pela região posicionada ao lado dos

itens alocados na parede, como ilustrado na Figura 4.9 (C), dado pela diferença da largura do espaço livre (esp^{livre}) pela largura ocupada pelos itens em $tamanho^{parede}$.

O espaço frontal ($esp^{frontal}$) é considerado pela região situada a frente dos itens alocados na parede, como representado na Figura 4.9 (D), dado pela diferença do comprimento do espaço livre (esp^{livre}) e o comprimento ocupado pelos itens em $tamanho^{parede}$.

Na Figura 4.11 é possível observar o diagrama que destaca a relação entre os algoritmos 1 a 7, descrevendo o funcionamento do método heurístico proposto no Capítulo 4, seções 4.1 e 4.2.

Figura 4.11: Algoritmos da Heurística.



Fonte: o Autor

5 EXPERIMENTOS COMPUTACIONAIS

Neste capítulo são apresentados os experimentos computacionais realizados a partir da heurística construtiva proposta na Capítulo 4 para o problema de carregamento de veículo considerado neste trabalho. Os procedimentos foram implementados em linguagem de programação C e executados em um computador com sistema operacional *Windows* 11 Pro v. 22H2, de 64 *bits*, processador Intel Core I5-8250U, 1.6 GHz e 12 GB de memória DDR4 SDRAM. Os modelos de Beasley (1985) e do problema da mochila foram resolvidos pelo *software* IBM ILOG CPLEX v. 22.1.1 com a configuração padrão. A fim de comparação, os tempos computacionais apresentados por Ranck Jr. (2014) foram obtidos em um computador com processador Intel Core i7-2860QM, 3.6 GHz com 8 GB de memória DDR3 SDRAM, em uma plataforma de 64 *bits*.

5.1 INSTÂNCIAS

Para os testes computacionais, foram utilizadas exemplares a partir das instâncias apresentadas em Ranck Jr. (2014) e Ranck Jr. et al. (2019), baseadas em um exemplo prático.

Os parâmetros considerados nos experimentos computacionais são:

- $g^{vx} = L/2$ e $g^{vy} = W/2$, o centro de gravidade almejado para o carregamento dos itens no contêiner, definido em seu centro geométrico, em relação aos eixos x e y , respectivamente;
- $\alpha = 1$, parâmetro para estabilidade vertical. Para $\alpha = 1$, indica-se que a face inferior de um item precisa estar 100% apoiada na face superior de outros itens, camadas ou na base do compartimento;
- $\beta = 0$ e $\gamma = 0$, parâmetros para estabilidade horizontal. Para $\beta = \gamma = 0$, não é imposta uma área mínima de contato das faces laterais de outros itens ou as laterais do compartimento;
- $\mu_c = 0,7 \cdot h_c$, limitante superior para o espaço destinado ao empacotamento das camadas completas no compartimento $c \in C$, em relação a altura h_c do compartimento;
- $\varepsilon^x = P_1^\Delta / |C| \cdot 0,02 \cdot L$ e $\varepsilon^y = P_1^\Delta / |C| \cdot 0,02 \cdot W$, tolerância para os desvios de balanceamento para os eixos x e y , respectivamente;
- $\bar{\psi}_j = 4$, $j \in N'$, quantidade mínima de itens de uma camada completa $j \in N'$ necessários para garantir a estabilidade dos itens acomodados acima dela. As posições dos itens na camada podem ser considerados nas extremidades desta camada;

- $\omega^{zp} = 0,2$ e $\omega^{zq} = 0,8$, a penalização para o remanejamento de itens, considerando 20% para o peso e 80% para a quantidade de itens;
- $\omega^z = 1$ e $\omega^{dev} = 1$, pesos relativos, respectivamente, para os fatores relacionados ao remanejamento de itens e os desvios de balanceamento de itens, para a função objetivo.

De acordo com os dados apresentados por Ranck Jr. (2014) em relação ao exemplo prático, o veículo considerado para o transporte apresenta uma divisão de 8 compartimentos, diferenciados por seus tamanhos: padrão (Pa) e rebaixado (Re), conforme apresentado na Tabela 5.1. O volume relativo (V^{rel}) representa a capacidade máxima transportada por um contêiner em relação a soma total dos volumes relativos dos itens $i \in M$ (v_i^{rel}) acomodados nos compartimentos $c \in C$. OS volumes relativos, conforme apresenta Ranck Jr. (2014), seguem o cálculo utilizado pela empresa, dado por $100/\max Q_i$, em que $\max Q_i$ é a quantidade máxima de itens do tipo i que a empresa coloca sobre um palete, com apenas este tipo de item.

Tabela 5.1: Informações do contêiner multicompartimentado.

Contêiner	V^{rel}	Compartimentos ($c \in C$)							
		1	2	3	4	5	6	7	8
8 compartimentos	760	Pa	Pa	Pa	Pa	Re	Re	Pa	Pa

Fonte: Ranck Jr. et al. (2019).

Na Tabela 5.2 são apresentadas as dimensões dos compartimentos, comprimento, largura e altura, para os dois tipos presentes no contêiner.

Tabela 5.2: Tipos de compartimentos e suas dimensões.

Tipo	Comprimento (L)	Largura (W)	Altura (H_c)
Padrão (Pa)	100	120	178
Rebaixado (Re)	100	120	148

Fonte: Ranck Jr. et al. (2019).

Um contêiner possui dimensões de comprimento $L = 2 \cdot L^{comp}$ e largura $W = (|C|/2) \cdot W^{comp}$, com $L, W \in \mathbb{Z}$. As posições de um compartimento c , seguindo o sistema de coordenadas definido para o contêiner, são dadas pelas coordenadas $(p_c, q_c, 0)$, sendo:

$$p_c = \begin{cases} L^{comp}, & \text{se } c \text{ é par} \\ 0, & \text{se } c \text{ é ímpar} \end{cases} \quad \text{e} \quad q_c = \left(\left\lceil \frac{c}{2} \right\rceil - 1 \right) \cdot W^{comp}$$

Em relação aos itens, os dados contemplam um universo com 32 tipos de itens, que apresentam características individuais como: comprimento (l_i), largura (w_i), altura (h_i), peso (p_i), com $l_i, w_i, h_i \in \mathbb{Z}$, pressão exercida (σ_i) e volume relativo (v_i^{rel}), apresentados na Tabela 5.3 e 5.4.

Para realização dos experimentos, foi considerado um contêiner com 8 compartimentos, seguindo os dados da Tabela 5.1. Consideramos inicialmente uma amostra de 13

tipos de itens com as respectivas demandas para 18 clientes, conforme pode ser observado na Tabela ???. A lista de paradas é pré-definida e considera a ordem em que os clientes devem ser visitados, ou seja, a primeira parada a receber entregas é o cliente 1, a segunda parada a receber entregas é o cliente 2 e assim sucessivamente. Em cada parada, a demanda deve ser entregue completamente, ou seja, todos os itens devem ter sido empacotados no processo de carregamento.

Tabela 5.3: Dados dos itens utilizados nos experimentos computacionais.

$i \in M$	l_i (cm)	w_i (cm)	h_i (cm)	p_i (kg)	σ_i (kg/cm ²)	v_i^{rel}
1	13	20	12	2.31	0.10	0.21
2	14	20	13	2.19	0.09	0.21
3	20	27	13	4.61	0.10	0.38
4	5	16	13	0.90	0.07	0.11
5	11	16	14	1.60	0.09	0.18
6	4	16	15	0.68	0.09	0.07
7	13	13	16	2.06	0.10	0.19
8	19	28	17	5.70	0.03	1.09
9	28	42	18	23.06	0.06	2.78
10	13	16	19	2.00	0.09	0.17
11	17	24	20	5.00	0.06	0.69
12	13	17	20	2.97	0.08	0.29
13	16	24	21	6.70	0.07	0.67
14	19	26	22	6.60	0.08	0.60
15	20	27	23	6.53	0.07	0.65
16	12	20	23	3.20	0.07	0.37
17	13	19	24	3.40	0.09	0.30
18	12	18	24	3.20	0.07	0.33
19	14	21	24	3.95	0.07	0.42
20	14	20	25	3.40	0.07	0.30
21	15	26	26	6.98	0.13	0.71
22	31	42	27	15.01	0.07	1.43
23	33	48	29	15.01	0.05	2.38
24	28	32	30	15.00	0.08	1.11
25	35	51	32	29.52	0.08	2.38
26	18	28	33	10.00	0.06	1.14
27	17	27	33	9.90	0.06	1.14
28	16	27	34	5.10	0.04	1.04

Continua na próxima página

Tabela 5.4 – Continuação da tabela 5.3

$i \in M$	l_i (cm)	w_i (cm)	h_i (cm)	p_i (kg)	σ_i (kg/cm ²)	v_i^{rel}
29	21	31	35	12.90	0.08	1.00
30	31	41	36	22.87	0.07	2.00
31	24	35	37	16.00	0.06	1.67
32	20	31	38	14.45	0.07	1.25

*Fim da tabela***Fonte:** Ranck Jr. et al. (2019).

A partir dos valores presentes na Tabela 5.3 e 5.4, um conjunto de 13 itens foram selecionados aleatoriamente e removidos da demanda total, para que a quantidade total de caixas seja 780, de modo a ficar condizente com o valor considerado por Ranck Jr. (2014). Entretanto, o peso total da carga não condiz com o valor encontrado por Ranck Jr. (2014), visto que foram retirados 13 itens da lista de demandas, sorteados aleatoriamente, para que as demandas totais fossem compatíveis.

Para os experimentos computacionais, foram gerados 10 exemplares considerados a partir do exemplo prático apresentado por Ranck Jr. (2014), que correspondem à Classe 1. Estas instâncias são obtidas a partir da troca da ordem de paradas apresentada na Tabela 5.5, mudando assim, a sequência em que os clientes devem ser visitados. Na Tabela 5.6 são apresentadas as sequências de paradas consideradas para cada uma das 10 instâncias geradas.

Tabela 5.5: Demandas dos tipos de itens por paradas.

Paradas (k)	Itens (i)												
	1	3	7	12	13	16	19	22	25	28	29	30	31
1	1	1	0	0	0	0	1	0	2	1	1	0	0
2	0	0	0	0	0	0	5	0	0	0	24	0	0
3	0	0	0	2	0	0	0	0	0	0	3	0	0
4	10	0	0	0	0	5	0	0	0	0	3	0	0
5	0	4	0	0	1	0	0	1	2	2	1	0	0
6	0	0	0	0	1	0	0	0	0	0	0	0	0
7	6	0	0	0	0	0	0	0	2	0	9	0	0
8	0	62	6	109	0	0	0	0	3	0	24	49	0
9	0	12	0	0	0	0	0	2	2	0	0	0	0
10	8	23	22	3	0	0	2	0	6	4	30	0	0
11	0	0	0	0	7	0	0	0	0	0	7	0	0
12	5	0	0	0	0	0	2	0	0	0	18	0	0
13	2	0	0	0	0	0	0	18	0	0	0	0	0
14	4	33	0	4	0	0	5	0	3	5	19	17	10
15	10	0	0	0	0	0	2	0	0	0	38	0	0
16	20	13	0	26	0	0	6	0	10	0	22	0	0
17	0	2	0	0	0	0	0	1	4	0	1	0	0
18	2	0	0	0	0	4	2	2	9	1	4	0	0

Fonte: Ranck Jr. et al. (2019).

Tabela 5.6: Sequências de paradas consideradas nas instâncias.

Instâncias	Paradas (k)																	
1I	1	11	3	13	5	14	6	15	7	18	9	4	17	12	7	2	15	10
2I	3	17	14	10	7	6	11	4	18	8	1	15	5	16	12	2	13	9
3I	1	13	6	18	11	4	16	9	3	14	7	17	10	12	5	15	7	2
4I	13	16	1	4	8	11	14	17	2	6	9	12	15	18	3	7	10	5
5I	6	3	18	17	14	11	7	4	15	12	8	5	1	16	9	2	10	13
6I	5	10	12	16	1	4	8	11	14	17	2	6	9	12	15	18	3	7
7I	3	9	16	4	11	17	5	12	18	6	13	1	8	14	2	15	7	10
8I	6	16	8	18	9	1	11	2	12	4	14	5	15	7	13	17	10	3
9I	8	11	14	17	2	5	9	12	15	18	3	6	7	10	13	16	1	4
10I	9	1	11	3	13	5	14	6	16	8	18	9	4	17	12	7	2	15

Fonte: o Autor.

5.2 RESULTADOS

O valor da função objetivo é analisado de acordo com o valor das penalidades para o remanejamento de itens durante o processo de entrega e os desvios de balanceamento no empacotamento da carga. Além disso, são apresentados os tempos computacionais das etapas de construção das camadas completas e das camadas incompletas. Também analisamos a quantidade total de itens carregados, a quantidade de camadas geradas, para ambos os tipos e porcentagem de itens residuais obtidos após a construção das camadas completas. O objetivo aqui é obter soluções factíveis, ao acomodar todos os itens demandados, e em menor tempo computacional, ao compararmos com os resultados apresentados por Ranck Jr. (2014), obtidos com a resolução de modelo matemáticos por métodos exatos.

Em relação a construção das camadas completas, o cálculo da quantidade total de itens em um camada é um dado conhecido e, portanto, resolvido na etapa de pré-processamento. Com isso, o tempo computacional deste processo não é contabilizado ao tempo de construção das camadas completas.

As informações analisadas na resolução dos experimentos são:

- cR : valor total das penalidades de remanejamento dos itens;
- cB : valor total das penalidades dos desvios de balanceamento;
- qR : quantidade total de itens remanejados;
- Tc : tempo computacional, em segundos, para gerar as camadas completas;
- Ti : tempo computacional, em segundos, para gerar as camadas incompletas;
- $|N'|$: número total de camadas completas;
- $|N''|$: número total de camadas incompletas;

Tabela 5.7: Informações da resolução dos exemplares 1I a 10I.

Instâncias	cR	cB	qR	Tc	Ti	$ N' $	$ N'' $	%res
1I	267.43	15572.98	94	0.193	0.003	36	4	25,26
2I	850.91	14344.23	268	0.202	0.005	36	4	25,26
3I	233.10	29732.92	75	0.249	0.004	37	4	24,62
4I	471.97	18274.32	160	0.206	0.003	37	4	24,62
5I	163.47	33502.04	61	0.259	0.004	36	4	25,26
6I	887.32	18857.90	284	0.193	0.002	37	4	24,62
7I	1096.52	29421.72	337	0.187	0.002	35	4	25,64
8I	499.66	9451.13	208	0.221	0.003	35	4	25,64
9I	121.66	11355.13	184	0.207	0,002	35	4	25,64
10I	190.76	17139.56	114	0.268	0.005	37	4	24,62
Mín.	121.66	9451.13	61	0.187	0.002	35	4	24,32
Máx.	1096.52	33502.90	337	0.268	0.005	37	4	25,64
Méd.	498.39	21834.00	159,98	0.2175	0.0033	36.3	4	25.016

Fonte: o Autor.

- %res: porcentagem de itens residuais em relação a demanda total;

Pode-se observar na Tabela 5.7 que foi possível obter uma solução factível em todos os exemplares, para a quantidade de 780 itens, ou seja, a demanda de todos os clientes foi atendida em suas totalidades. Nos resultados obtidos nas instâncias geradas, pode-se notar a influência que a sequência de entregas exerce sobre os valores dos custos de remanejamento e dos desvios de balanceamento da carga, em que a ordem de paradas pode levar a custos mais altos ou mais baixos. Com isso, vale destacar a importância de determinar uma sequência de entregas eficiente, para que seja possível alcançar custos mais baixos.

Observando os resultados destacados na Tabela 5.7, em relação ao valores apresentados por Ranck Jr. (2014), o custo médio obtido para o remanejamento dos itens, foi de 498.39, revelando uma consistência notável em relação ao valor anteriormente reportado, que foi de 444.89. No que diz respeito aos desvios de balanceamento, o custo médio alcançado foi de 21834.00, estando próximo do valor de referência 17345,89 apresentado por Ranck Jr. (2014).

No processo de carregamento, ficou visível a relevância de acomodar o máximo de itens possível em camadas completas, para que o atendimento total das demandas fosse atingido. Para isso, a estratégia de utilizar o cálculo da quantidade máxima de caixas, para cada tipo de item $i \in M$, nas camadas completas é de grande importância. Ademais, adotar o carregamento completo de alguns compartimentos, determinando uma sequência de camadas completas que preenchessem totalmente o espaço livre, contribuiu para o empacotamento de todos os itens residuais. Estas etapas levaram a obter um conjunto de itens residuais com uma média de 25.016% dos itens demandados, valor este próximo a média obtida por Ranck Jr. (2014) de 21,15%, indicando uma proximidade em termos de escala de grandeza.

Em relação aos tempos computacionais apresentados Ranck Jr. (2014), em

todas as instâncias da Classe 1, o empacotamento dos itens realizado pela resolução do modelo de programação linear inteira mista atingiu o tempo limite de 1200 segundos. O tempo médio para resolução deste modelo foi de 1200,68 e para a construção das camadas incompletas, o tempo médio foi de 1.32 segundos. A utilização do modelo para resolução do empacotamento de itens foi o principal fator que acarretou no aumento do tempo de resolução do problema, em que o tempo médio total de resolução foi de 1217.74 segundos. Com a heurística construtiva, o tempo médio observado na construção das camadas incompletas foi de 0.0033 segundos. Em relação ao processo de empacotamento, constatou-se um tempo médio de 0.2208 segundos, considerando as etapas destinadas a gerar as camadas completas e as camadas incompletas.

Para etapa de pré-processamento, considerando os tipos de itens listados na Tabela 5.4, a aplicação do modelo de Beasley (1985) para calcular a quantidade máxima de itens em uma camada completa, de maneira geral, resultou em tempos de execução inferiores a 1 segundo para a maioria dos casos. Entretanto, para os itens dos tipos 1, 4, 5, 6, 10 e 12, o tempo de resolução foi superior a este valor, com destaque para o item do tipo 4, que apresentou sua resolução com um tempo de 235,331 segundos. No Anexo A são apresentados os padrões de empacotamento gerados pela resolução do modelo para cada tipo de item.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Neste trabalho foram apresentadas uma revisão da tipologia proposta por Wäscher, Haußner e Schumann (2007) para os problemas de corte e empacotamento, a classificação das restrições práticas de problemas de empacotamento descritas em Bortfeldt e Wäscher (2013) e a tipologia apresentada por Ostermeier et al. (2021) para os problemas de roteamento de veículos multicompartimentados. A revisão destes possibilitou estabelecer um entendimento teórico relevante dos estudos desenvolvidos acerca de problemas de corte e empacotamento, mais especificamente, dos problemas de carregamento de veículos multicompartimentados e as principais áreas de aplicação consideradas na literatura.

Como aplicação, abordamos um problema relacionado ao carregamento de veículo multicompartimentado, inicialmente discutido por Ranck Jr. (2014). Este cenário foi observado em uma empresa da indústria de bebidas, onde caminhões compartimentados eram utilizados para a distribuição de produtos. Para isso, foi apresentada uma heurística construtiva como proposta de resolução para este problema, que envolve o empacotamento eficiente dos itens nos compartimentos do veículo. A complexidade do problema reside na necessidade de distribuir os itens em camadas horizontais, dos tipos completas e incompletas, a fim de otimizar o espaço disponível no veículo.

O desenvolvimento apresentado na literatura para este problema considera um conjunto de instâncias geradas a partir de um exemplo prático de uma empresa de bebidas. A solução proposta utiliza modelos de Programação Linear Inteira Mista, que podem acarretar em um alto tempo computacional de resolução. Em virtude disso, o objetivo deste trabalho foi desenvolver um método de resolução capaz de encontrar uma solução factível, empacotando todos os itens nos compartimentos, em menor tempo computacional de execução.

Na construção das camadas completas, esforços foram dedicados para o cálculo da quantidade máxima de caixas e suas orientações nestas estruturas para cada tipo de item, resolvendo esta etapa como um problema de empacotamento bidimensional. Este processo foi considerado como um pré-processamento, a ser resolvido antes do processo de carregamento. O empacotamento das camadas completas ocorre pela distribuição dos itens nas camadas, seguindo a lista de entregas na ordem inversa de paradas. Para construção das camadas incompletas, consideramos a distribuição dos itens em paredes verticais a partir da construção de blocos de itens. Os blocos são formados por pilhas de itens, considerando o espaço livre disponível para construção das camadas incompletas em cada um dos compartimentos com espaço reservado para estas estruturas. Este processo foi resolvido como um problema de empacotamento tridimensional com os itens residuais.

Como podemos observar com os experimentos computacionais, a utilização do método heurístico possibilitou atingir soluções factíveis para todos os casos estudados, atendendo plenamente à demanda total de 780 itens. É importante destacar que a sequência de

entregas influencia nos custos de remanejamento e desvios de balanceamento, enfatizando a necessidade de determinar de maneira eficiente a sequência de entregas, a fim de alcançar uma redução dos custos. Pode-se observar também que, para o atendimento total das demandas e obter um carregamento otimizado, foi essencial priorizar a distribuição dos itens em camadas completas. Além disso, a heurística construtiva demonstrou-se eficaz, apresentando tempos de resolução satisfatórios e viável para a otimização do carregamento de veículos multicompartimentados.

Possíveis melhorias podem ser desenvolvidas em trabalhos futuros. Entre estas, encontra-se a análise de sensibilidade entre os desvios de remanejamento e os desvios de balanceamento. Para resolver o *subset-sum problem* durante o processo de empacotamento, foi utilizado um software comercial de otimização e, para reduzir ainda mais o tempo computacional, poderia ser utilizado um método desenvolvido especificamente para o problema.

Como o método proposto tem um baixo custo computacional, também é possível estender o problema estudado para resolver o problema de empacotamento com múltiplos veículos multicompartimentados. Neste caso, o método heurístico pode ser utilizado para resolver os subproblemas do problema maior. Ademais, é possível modificar a heurística construtiva implementada inicialmente para uma metaheurística GRASP (Greedy Randomized Adaptive Search Procedure), a fim de obter um carregamento mais eficiente. A metaheurística GRASP é um método iterativo composto de duas etapas, uma fase construtiva e uma fase de busca local. Uma interessante pesquisa futura é considerar a utilização de diferentes procedimentos de busca local. Como os algoritmos VND (*Variable Neighborhood Descent*) e VNS (*Variable Neighborhood Search*), que podem refinar a abordagem GRASP, gerando soluções de melhor qualidade.

REFERÊNCIAS

- BAUSCH, D. O.; BROWN, G. G.; RONEN, D. Scheduling short-term marine transport of bulk products. *Maritime Policy & Management*, Taylor & Francis, v. 25, n. 4, p. 335–348, 1998.
- BEASLEY, J. E. An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research*, INFORMS, v. 33, n. 1, p. 49–64, 1985.
- BEZERRA, V. M. et al. Models for the two-dimensional level strip packing problem—a review and a computational evaluation. *Journal of the Operational Research Society*, Taylor & Francis, v. 71, n. 4, p. 606–627, 2020.
- BEZERRA, V. M. R. *Problemas de empacotamento bidimensional em níveis: estratégias baseadas em modelagem matemática*. Tese (Doutorado) — Universidade de São Paulo, 2018.
- BORGES, Y. G. et al. Exact algorithms for class-constrained packing problems. *Computers & Industrial Engineering*, Elsevier, v. 144, p. 106455, 2020.
- BORTFELDT, A.; WÄSCHER, G. Constraints in container loading—a state-of-the-art review. *European Journal of Operational Research*, Elsevier, v. 229, n. 1, p. 1–20, 2013.
- BROWN, G. G. et al. Real-time, wide area dispatch of mobil tank trucks. *Interfaces*, INFORMS, v. 17, n. 1, p. 107–120, 1987.
- CHANG, P. The vehicle routing optimization problem of refrigerated vehicle with temperature compartments under consistent delivery model. *Academic Journal of Science and Technology*, v. 1, n. 2, p. 1–9, 2022.
- CHRISTOFIDES, N.; WHITLOCK, C. An algorithm for two-dimensional cutting problems. *Operations Research*, INFORMS, v. 25, n. 1, p. 30–44, 1977.
- CORNILLIER, F. et al. A heuristic for the multi-period petrol station replenishment problem. *European Journal of Operational Research*, Elsevier, v. 191, n. 2, p. 295–305, 2008.
- DERIGS, U. et al. Vehicle routing with compartments: applications, modelling and heuristics. *OR spectrum*, Springer, v. 33, p. 885–914, 2011.
- DYCKHOFF, H.; FINKE, U. *Cutting and packing in production and distribution: A typology and bibliography*. [S.l.]: Springer Science & Business Media, 1992.
- EGLESE, R. W.; MERCER, A.; SOHRABI, B. The grocery superstore vehicle scheduling problem. *Journal of the Operational Research Society*, Springer, v. 56, p. 902–911, 2005.
- ELBEK, M.; WØHLK, S. A variable neighborhood search for the multi-period collection of recyclable materials. *European Journal of Operational Research*, Elsevier, v. 249, n. 2, p. 540–550, 2016.
- GAJDA, M. et al. An optimization approach for a complex real-life container loading problem. *Omega*, Elsevier, v. 107, p. 102559, 2022.

- GHIDINI, C. T. L. d. S. *Otimização de processos acoplados: programação da produção e corte de estoque*. Tese (Doutorado) — Universidade de São Paulo, 2009.
- GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem. *Operations research*, INFORMS, v. 9, n. 6, p. 849–859, 1961.
- GILMORE, P. C.; GOMORY, R. E. Multistage cutting stock problems of two and more dimensions. *Operations research*, INFORMS, v. 13, n. 1, p. 94–120, 1965.
- HERZ, J. Recursive computational procedure for two-dimensional stock cutting. *IBM Journal of Research and Development*, IBM, v. 16, n. 5, p. 462–469, 1972.
- HÜBNER, A.; OSTERMEIER, M. A multi-compartment vehicle routing problem with loading and unloading costs. *Transportation Science*, INFORMS, v. 53, n. 1, p. 282–300, 2018.
- JUNQUEIRA, L. *Modelos de programação matemática para problemas de carregamento de caixas dentro de contêineres*. Dissertação (Mestrado) — Universidade Federal de São Carlos, 2009.
- KALLRATH, J. et al. Solving real-world cutting stock-problems in the paper industry: Mathematical approaches, experience and challenges. *European Journal of Operational Research*, Elsevier, v. 238, n. 1, p. 374–389, 2014.
- KANTOROVICH, L. V. Mathematical methods of organizing and planning production. *Management science*, INFORMS, v. 6, n. 4, p. 366–422, 1960.
- LEÃO, A. A. et al. The constrained compartmentalized knapsack problem: mathematical models and solution methods. *European journal of operational research*, Elsevier, v. 212, n. 3, p. 455–463, 2011.
- LEÃO, A. A. S. *Extensões em problemas de corte: padrões compartimentados e problemas acoplados*. Tese (Doutorado) — Universidade de São Paulo, 2013.
- LODI, A.; MARTELLO, S.; MONACI, M. Two-dimensional packing problems: A survey. *European journal of operational research*, Elsevier, v. 141, n. 2, p. 241–252, 2002.
- MARTINS, S. et al. Product-oriented time window assignment for a multi-compartment vehicle routing problem. *European Journal of Operational Research*, Elsevier, v. 276, n. 3, p. 893–909, 2019.
- MOURA, A.; OLIVEIRA, J. F. A grasp approach to the container-loading problem. *IEEE Intelligent Systems*, IEEE, v. 20, n. 4, p. 50–57, 2005.
- MUYLDERMANS, L.; PANG, G. On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. *European Journal of Operational Research*, Elsevier, v. 206, n. 1, p. 93–103, 2010.
- OPPEN, J.; LØKKETANGEN, A. A tabu search approach for the livestock collection problem. *Computers & Operations Research*, Elsevier, v. 35, n. 10, p. 3213–3229, 2008.
- OSTERMEIER, M. et al. Multi-compartment vehicle routing problems: State-of-the-art, modeling framework and future directions. *European Journal of Operational Research*, Elsevier, v. 292, n. 3, p. 799–817, 2021.

OSTERMEIER, M. et al. Loading constraints for a multi-compartment vehicle routing problem. *Or Spectrum*, Springer, v. 40, n. 4, p. 997–1027, 2018.

PIRKWIESER, S.; RAIDL, G. R.; GOTTLIEB, J. Improved packing and routing of vehicles with compartments. In: SPRINGER. *International Conference on Computer Aided Systems Theory*. [S.l.], 2011. p. 392–399.

POLDI, K. C. *Algumas extensões do problema de corte de estoque*. Tese (Doutorado) — Universidade de São Paulo, 2003.

QUIROGA-OROZCO, J. J.; CARVALHO, J. M. Valério de; HOTO, R. S. V. A strong integer linear optimization model to the compartmentalized knapsack problem. *International Transactions in Operational Research*, Wiley Online Library, v. 26, n. 5, p. 1633–1654, 2019.

RANCK Jr., R. *Contribuições para um problema de empacotamento em um veículo multicompartimentado*. Tese (Doutorado) — São José dos Campos: INPE,, 2014.

RANCK Jr., R. et al. A hybrid approach for a multi-compartment container loading problem. *Expert Systems with Applications*, Elsevier, v. 137, p. 471–492, 2019.

RUIZ, R.; MAROTO, C.; ALCARAZ, J. A decision support system for a real vehicle routing problem. *European Journal of Operational Research*, Elsevier, v. 153, n. 3, p. 593–606, 2004.

SILVA, E.; OLIVEIRA, J. F.; WÄSCHER, G. The pallet loading problem: a review of solution methods and computational experiments. *International Transactions in Operational Research*, Wiley Online Library, v. 23, n. 1-2, p. 147–172, 2016.

SILVA, E. F.; TOFFOLO, T. A. M.; WAUTERS, T. Exact methods for three-dimensional cutting and packing: A comparative study concerning single container problems. *Computers & Operations Research*, Elsevier, v. 109, p. 12–27, 2019.

SILVA, E. F. da et al. A matheuristic framework for the three-dimensional single large object placement problem with practical constraints. *Computers & Operations Research*, Elsevier, v. 124, p. 105058, 2020.

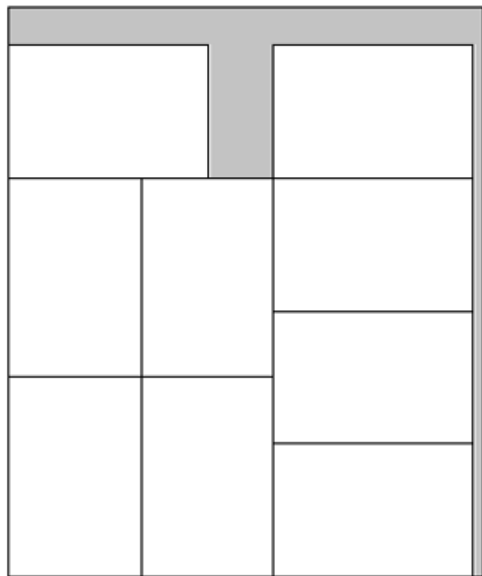
SISWANTO, N.; ESSAM, D.; SARKER, R. Solving the ship inventory routing and scheduling problem with undedicated compartments. *Computers & Industrial Engineering*, Elsevier, v. 61, n. 2, p. 289–299, 2011.

STADTLER, H. A one-dimensional cutting stock problem in the aluminium industry and its solution. *European Journal of Operational Research*, Elsevier, v. 44, n. 2, p. 209–223, 1990.

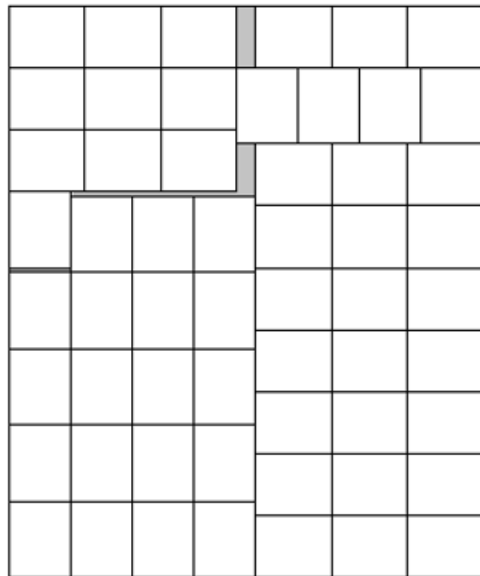
WÄSCHER, G.; HAUSSNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. *European journal of operational research*, Elsevier, v. 183, n. 3, p. 1109–1130, 2007.

ZHU, W.; LIM, A. A new iterative-doubling greedy–lookahead algorithm for the single container loading problem. *European Journal of Operational Research*, Elsevier, v. 222, n. 3, p. 408–417, 2012.

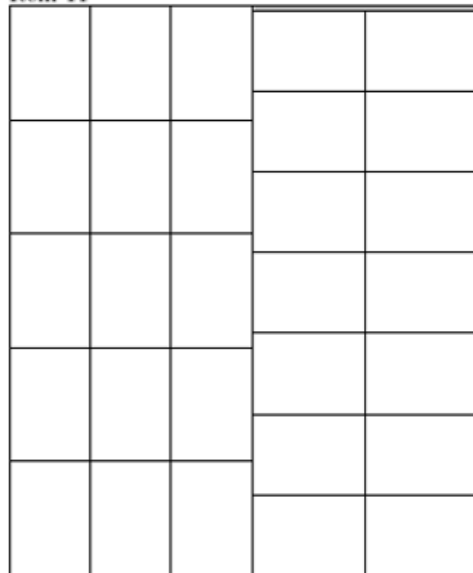
Item 9



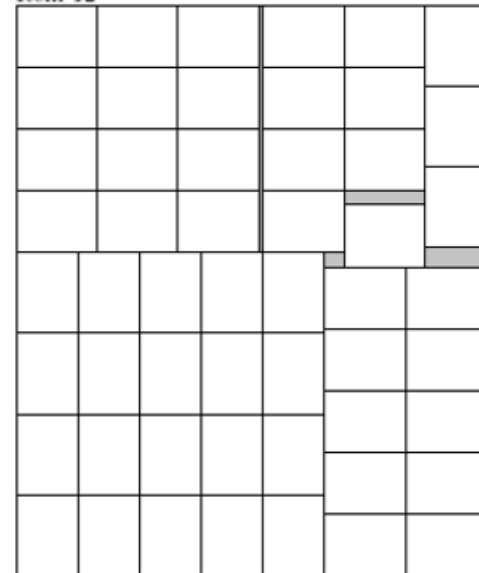
Item 10



Item 11



Item 12



Item 13

Item 14

Item 15

Item 16

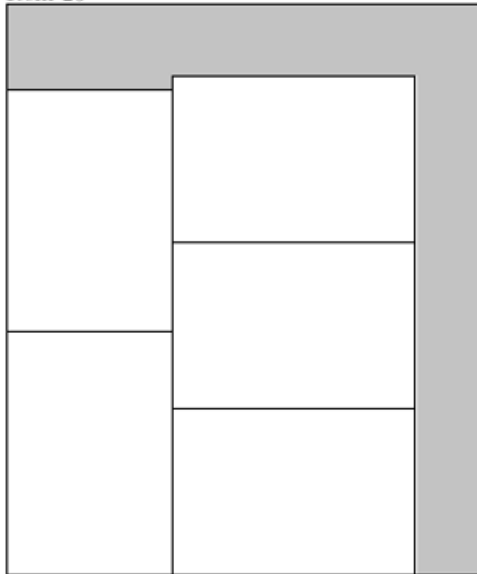
Item 21

Item 22

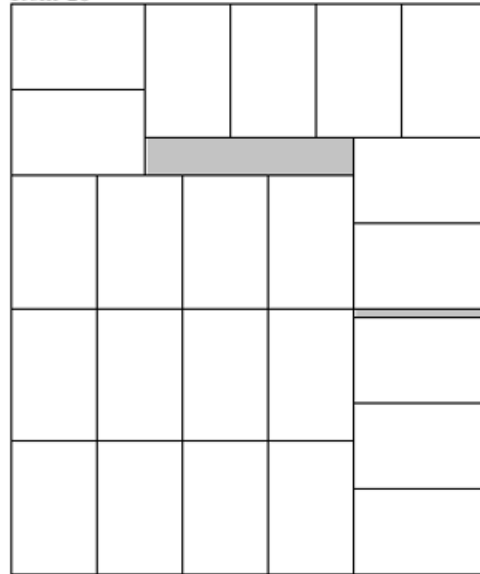
Item 23

Item 24

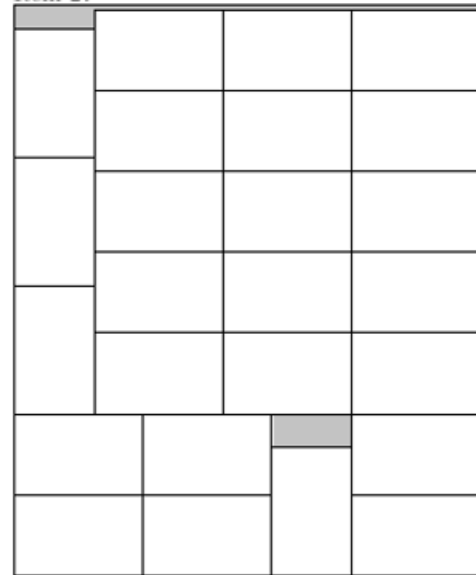
Item 25



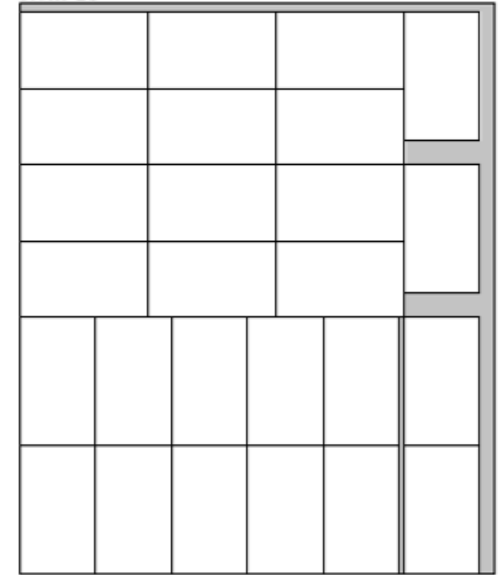
Item 26



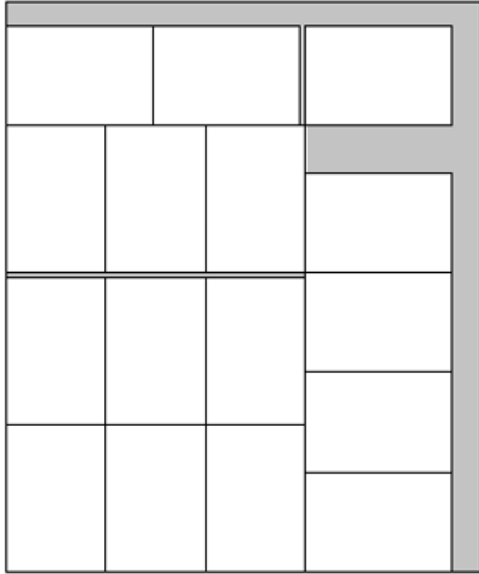
Item 27



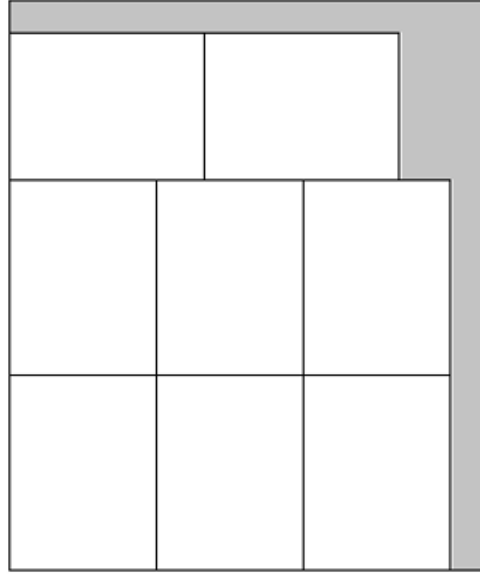
Item 28



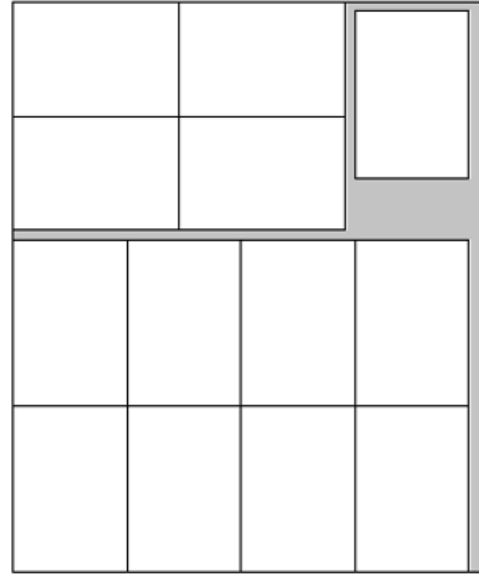
Item 29



Item 30



Item 31



Item 32

