

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL – UFRGS
ESCOLA DE ADMINISTRAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO

DANIEL BENTO MAIA

**RESOLUÇÃO DO PROBLEMA DO EMPACOTAMENTO
TRIDIMENSIONAL UTILIZANDO O ALGORITMO DE
OTIMIZAÇÃO DA COLÔNIA DE FORMIGAS E O MÉTODO
DE GERAÇÃO DE COLUNAS**

Porto Alegre

2023

Daniel Bento Maia

Resolução do Problema do Empacotamento Tridimensional
utilizando o Algoritmo de Otimização da Colônia de Formigas e o
Método Geração de Colunas

Dissertação apresentada como
requisito parcial para obtenção do grau
de Mestre em Administração da
Universidade Federal do Rio Grande do
Sul - UFRGS, Escola de Administração.

Orientador: Prof. Dr. Denis Borenstein

Data de Aprovação:

Banca Examinadora

Prof. Dr. Denis Borenstein
Universidade Federal do Rio Grande do Sul
Orientador

Prof. Dr. Olinto César Bassi de Araújo
Universidade Federal de Santa Maria
Examinador

Prof. Dr. Luciano Ferreira
Universidade Federal do Rio Grande do Sul
Examinador

Prof. Dr. Pablo Cristini Guedes
Universidade Federal do Rio Grande do Sul
Examinador

Maia, Daniel Bento
Resolução do Problema do Empacotamento
Tridimensional utilizando o Algoritmo de Otimização da
Colônia de Formigas e o Método Geração de Colunas /
Daniel Bento Maia. -- 2023.
63 f.
Orientador: Denis Borenstein.

Dissertação (Mestrado) -- Universidade Federal do
Rio Grande do Sul, Escola de Administração, Programa
de Pós-Graduação em Administração, Porto Alegre,
BR-RS, 2023.

1. Problema do Empacotamento Tridimensional. 2.
Problema de Carregamento. 3. Heurística. 4. Algoritmo
de Otimização da Colônia de Formigas. 5. Método de
Geração de Colunas. I. Borenstein, Denis, orient. II.
Título.

RESUMO

O presente trabalho tem como objetivo apresentar um método de solução para o Problema do Empacotamento Tridimensional com rotação. Este problema apresenta uma aplicação prática no carregamento de contêineres, caminhões, depósitos e no corte de estoque tridimensional não guilhotinado. Por se tratar de um problema *NP-Hard*, muitas vezes não é possível encontrar sua solução ótima em tempo razoável. A resolução proposta é a utilização do Algoritmo de Otimização da Colônia de Formigas combinado com o Método de Geração de Colunas para a resolução do problema. Testes com instâncias da literatura demonstram que esta meta-heurística possui potencial para resolver o problema com eficiência e eficácia.

Palavras chaves: Problema do Empacotamento Tridimensional. Problema de Carregamento. Heurística. Algoritmo de Otimização da Colônia de Formigas. Método de Geração de Colunas.

ABSTRACT

This study aims to present solution method for Tridimensional Bin Packing Problem with rotation. This problem has a practical application in containers, truck and deposits loading and non-guillotined cutting-stock problems. In fact of problem been a NP-Hard, in many cases are not possible find the optimal solution in reasonable time. The proposal solution is the utilization of Ant Colony Optimization Algorithm combined with Column Generation Method to solve this problem. Tests with instances from the literature demonstrate that this metaheuristic has the potencial to solve the problem efficiently and effectively.

Key words: Tridimensional Bin Packing Problem. Loading Problem. Heuristic. Ant Colony Optimization. Column Generation Method.

LISTA DE ILUSTRAÇÕES

Figura 1 - Relação entre Problema do Empacotamento Tridimensional (a), Bidimensional (b) e Unidimensional (c).....	11
Figura 2 – Variáveis e Parâmetros PET	13
Figura 3 - Posição relativa entre itens	14
Figura 4 – Atualização do Espaço Residual	24
Figura 5 – Pontos Externos criados por um item	25
Figura 6 – Exemplo de Espaços Residuais	26
Figura 7 - Representação do caminho a ser percorrido pelas formigas	28
Figura 8 - Iterações entre diferentes partes da Técnica de Geração de Colunas.....	34
Gráfico 1 – Proporção de Formigas (Q) em instâncias com 50 itens	46
Gráfico 2 - Proporção de Formigas (Q) em instâncias com 150 itens	47
Gráfico 3 – Feromônio excretado (p) em instâncias com 50 itens	47
Gráfico 4 – Feromônio excretado (p) em instâncias com 150 itens.....	48
Gráfico 5 – Acréscimo por Qualidade (a) em instâncias com 50 itens	48
Gráfico 6 – Acréscimo por Qualidade (a) em instâncias com 150 itens	49

LISTA DE TABELAS

Tabela 1 – Orientações dos itens e projeções nos eixos	14
Tabela 2 - Relação entre os parâmetros do ACO e a convergência da solução.....	33
Tabela 3 – Intervalo de dimensões do item	42
Tabela 4 – Percentual de tipo de itens que compõe cada classe de problemas.....	42
Tabela 5 – Percentual de itens (por volume) que compõe cada classe de problemas	43
Tabela 6 – Comparação entre Desempenhos da Heurística de Pontos Extremos ...	44
Tabela 7 - Limite tempo de execução para o Método de Geração de Colunas	45
Tabela 8 - Valores de Parâmetros ACO (Classe 1 a 4)	49
Tabela 9 – Valores de Parâmetros ACO (Classe 5 a 8)	50
Tabela 10 - Comparação de desempenho entre os Métodos Implementados.....	51
Tabela 11 – Gap Médio do Valor da Função Objetivo em Relação ao GC por Classe de Problema	52
Tabela 12 – Gap Médio de Tempo de Execução em Relação ao CG por classe de problema	52
Tabela 13 – Gap Médio do Valor da Função Objetivo em Relação ao CG por quantidade de itens	53
Tabela 14 – Gap Médio de Tempo de Execução em Relação ao CG por quantidade de itens.....	53
Tabela 15 – Comparação de Desempenho com o Estado da Arte	54
Tabela 16 –Gap Médio do Valor da Função Objetivo para o Estado da Arte por classe de problema	55
Tabela 17 – Gap Médio de Tempo de Execução para o Estado da Arte por classe de problema	56
Tabela 18 – Gap Médio do Valor da Função Objetivo para o Estado da Arte por quantidade de itens	56
Tabela 19 – Gap Médio de Tempo de Execução em Relação ao Estado da Arte por quantidade de itens	56

SUMÁRIO

1 INTRODUÇÃO	8
2 REFERENCIAL TEÓRICO	10
2.1 Problema do Empacotamento Tridimensional.....	10
2.2 Base para modelagem do PET	13
2.2 Modelagem do Problema de Maximização de Saídas	15
2.3 Modelagem do Problema de Minimização de Entradas	17
2.5 Métodos Heurísticos de Solução.....	19
3 METODOLOGIA.....	22
3.1 Formulação do problema	22
3.2 Modelagem do Problema	22
3.2.1 Heurística de Pontos Extremos.....	23
3.2.2 O Algoritmo de Otimização da Colônia de Formigas	27
3.2.3 Método de Geração de Colunas	33
3.2.4 Algoritmo de Otimização da Colônia de Formigas e Método de Geração de Colunas	38
3.2.5 Método de Geração de Colunas e Algoritmo de Otimização da Colônia de Formigas	38
4 TESTES COMPUTACIONAIS.....	42
4.1 Heurística de Pontos Extremos.....	43
4.2 Método de Geração de Colunas	45
4.3 Algoritmo de Otimização da Colônia de Formigas	45
4.4 Comparação entre as implementações.....	50
4.5 Comparação com Estado da Arte	53
5 CONCLUSÃO.....	58
BIBLIOGRAFIA	60

1 INTRODUÇÃO

O Problema do Empacotamento Tridimensional (PET) originou-se a partir do Problema do Corte de Estoque Tridimensional (GILMORE e GOMORY, 1965) e consiste em acomodar ortogonalmente um determinado número de itens de formato de paralelepípedo retângulo dentro de caixas ou compartimentos também em formato de paralelepípedo retângulo. O objetivo pode ser minimizar a quantidade de compartimentos utilizados ou maximizar a quantidade de itens carregados, com a condição de que os itens não ultrapassem os limites do compartimento ou se sobreponham entre si.

O Problema do Empacotamento Tridimensional tem várias aplicações práticas, entre elas, carregamento de itens para transporte, seja em armazéns, contêineres, vagões, caminhões ou paletes, desde que, tanto o compartimento quanto os itens a serem acomodados, sejam retangulares. O crescimento do transporte marítimo (ALI et. al., 2022) e a explosão dos preços dos contêineres marítimos (CONFEDERAÇÃO NACIONAL DA INDÚSTRIA, 2021) demonstram a importância prática do PET.

O PET pode ter diversas características, podendo variar quanto a quantidade de compartimentos, que pode ser apenas um ou vários; quanto a dimensão dos compartimentos, que pode ser única ou heterogênea; ou quanto às dimensões dos itens, que podem ser idênticos, fracamente heterogêneos ou fortemente heterogêneos (KURPEL, 2015). Também os itens a serem empacotados podem ser rotacionados ou não, dependendo da configuração do problema.

A resolução de grandes instâncias em sua otimalidade pode não ser viável por se tratar de um problema *NP*-Difícil. Embora alguns métodos heurísticos tenham sido aplicados na resolução do problema, eles também apresentam dificuldades em solucionar grandes instâncias em tempo razoável.

O presente trabalho tem como objetivo apresentar um método para resolução do PET com rotação em que o objetivo é minimizar a quantidade de compartimentos utilizados, encaixando-se no problema de minimização de entradas, como definiremos posteriormente. Para este estudo, os compartimentos são homogêneos, ou seja, todos tem as mesmas dimensões e os itens a serem inseridos

são fortemente heterogêneos. Trata-se de um problema teórico, pois não apresenta outras restrições práticas além de todos os itens serem carregados sem ultrapassarem os limites dos compartimentos e sem haver sobreposição entre os itens.

O método desenvolvido integra o Algoritmo de Otimização da Colônia de Formigas (ACO) e Método de Geração de Colunas (GC) para a resolução do Problema do Empacotamento Tridimensional com rotação. O ACO, no método proposto, é responsável por guiar a sequência de empacotamento dos itens através de uma heurística construtiva conhecida como Pontos Extremos. A incrementação do Algoritmo da Colônia de Formigas à heurística de Pontos Extremos permite a exploração do espaço solução através da diversificação (exploração de novas direções) e intensificação (aprofundar mais em direções que se mostraram promissoras). Já o método de Geração de Colunas permite a combinação das diversas soluções encontradas pelo ACO, o que pode resultar em uma solução melhor que as obtidas na fase anterior. Também foi demonstrado que pode se utilizar o método de Geração de Colunas para se chegar a uma boa solução e utilizar o ACO para tentar melhorar essa solução.

Três variantes do método foram utilizadas e aplicadas às instâncias criadas por Martello et. al. (2000). A primeira foi utilizando apenas o Algoritmo de Otimização da Colônia de Formigas, a segunda variante cria soluções com o ACO e as combinam utilizando o Método de Geração de Colunas (ACO + CG). A terceira variante do método, cria uma solução com a GC e busca o aprimoramento desta solução através do ACO (CG + ACO).

Este documento está organizado em mais 4 seções. Na próxima seção encontra-se uma da revisão literatura abordando o Problema do Empacotamento de uma forma geral, demonstrando 2 modelos de Programação Linear Inteira e citando alguns métodos heurísticos utilizados na resolução do PET. Na seção 3 está a apresentação e detalhamento dos métodos desenvolvidos. Na seção 4 estão registrados os testes computacionais e a comparação com o Estado da Arte e na última seção são apresentadas as conclusões.

2 REFERENCIAL TEÓRICO

2.1 PROBLEMA DO EMPACOTAMENTO TRIDIMENSIONAL

O Problema do Empacotamento apresenta grande similaridade com o Problema de Corte de Estoque, sendo ambos pertencentes à Otimização Combinatória. O Problema do Empacotamento consiste em acomodar objetos menores dentro de compartimentos maiores, enquanto o Problema de Corte de Estoque consiste em cortar peças menores a partir de unidades maiores. Estes problemas são considerados duais, já que um Problema do Empacotamento pode ser formulado como um Problema de Corte de Estoque, e vice e versa (FANSLAU e BORTFELDT, 2010). De fato, no primeiro problema, o espaço ocupado por um item corresponde à parte do material que será cortado para a produção de um item no segundo problema (ARENALES et. al., 2004).

De acordo com a dimensão, tanto o Problema do Empacotamento quanto o Problema de Corte de Estoque podem ser classificados em unidimensional, bidimensional ou tridimensional. No problema unidimensional (PEU) dado itens de tamanho w_i , tal que $0 < w_i < W$, e recipientes com capacidade W , o objetivo é empacotar os itens no menor número de recipientes (LARA e LABRADOR, 2010).

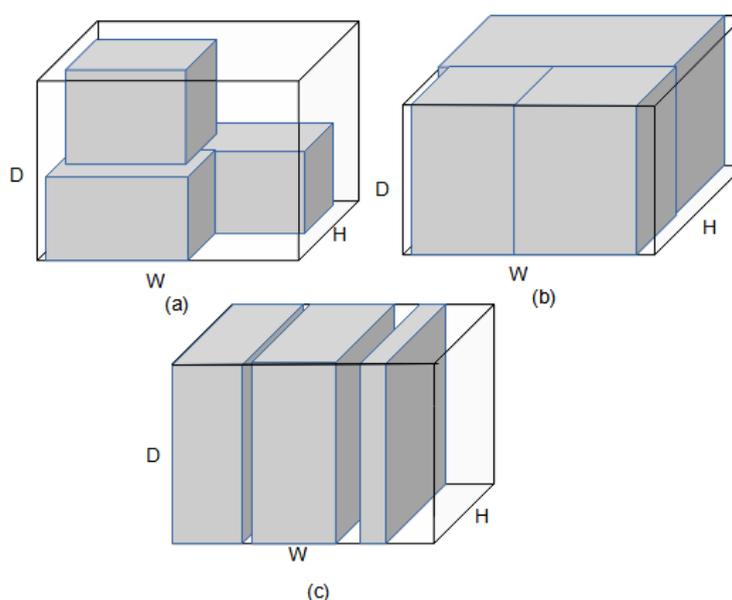
O Problema do Empacotamento Bidimensional (PEB) leva em consideração duas dimensões e pode ser utilizado quando, por exemplo, não se pode empilhar os itens e, desta forma, a altura não é uma dimensão relevante do problema. A partir de duas dimensões, não é mais suficiente apenas fazer a atribuição de cada item a seu compartimento, mas passa a ser necessário também indicar a posição de cada item e sua respectiva orientação.

Além de adicionar mais uma dimensão, o Problema do Empacotamento Tridimensional cresce em complexidade ao aumentar de 2 para 6 as possibilidades de orientação dos itens.

O Problema do Empacotamento Unidimensional é *NP*-Difícil, o que é provado pela redução do Problema da Mochila, conhecidamente pertencente aos Problemas *NP*-Difíceis (SCHEITHAUER, 1991). Os Problemas de Empacotamento Bidimensional e Tridimensional são *NP*-Difíceis, o que pode ser provado pela redução do Problema do Empacotamento Unidimensional (EGEBLAD e PISINGER,

2009). Dadas 3 dimensões de itens (w, h, d) e 3 dimensões de compartimentos (W, H, D) , Martello et. al. (2000) demonstram que o Problema do Empacotamento Tridimensional em que $d = D$ para todos os itens (Figura 1-b) é um Problema do Empacotamento Bidimensional. Já o Problema do Empacotamento Bidimensional em que $h = H$ para todos os itens (Figura 1-c) é um Problema do Empacotamento Unidimensional.

Figura 1 - Relação entre Problema do Empacotamento Tridimensional (a), Bidimensional (b) e Unidimensional (c)



Fonte: elaborada pelo autor

Ali et. al. (2022) divide de forma ampla os Problemas de Empacotamento Tridimensional em *off-line* e *on-line*. O PET *off-line* é quando se tem com antecedência as informações da quantidade e dimensões dos itens a serem empacotados. No PET *on-line*, os itens vão sendo carregados um a um, na ordem em que vão aparecendo, sem um conhecimento prévio de suas dimensões ou da quantidade total de itens.

Segundo Kurpel et. al. (2020), os Problemas de Corte e Empacotamento também podem ser divididos em 2 grupos de problemas: de minimização de entradas e de maximização de saídas. Os problemas de minimização de entradas são problemas em que os compartimentos são suficientes para empacotar todos os itens, neste caso, o objetivo é utilizar a menor quantidade possível de

compartimentos. Os problemas de maximização de saída são os que o objetivo é empacotar o máximo de itens possível, já que os compartimentos não são suficientes para empacotar todos os itens. Kurpel et. al. (2020) ainda subdividem cada grupo em 7 categorias conforme as características dos compartimentos (único, de tamanho homogêneo ou de tamanho heterogêneo) e as características dos itens (único, de tamanho homogêneo ou de tamanho heterogêneo).

Na resolução de problemas do mundo real, geralmente é necessário o incremento de restrições práticas ao modelo. Nascimento et. al. (2021) apresentam 4 restrições práticas para os problemas de maximização de saídas:

- I. carregamento completo (conjunto de itens que, ou todos são carregados, ou não se carrega nenhum);
- II. conflito de itens (itens que não podem ser carregados juntos);
- III. prioridade de carregamento; e
- IV. limite de peso.

Já as restrições para os problemas de minimização de entradas referem-se:

- I. a estabilidade da carga;
- II. o peso suportado pelos itens;
- III. a organização dos itens de acordo com o descarregamento em cada local de parada;
- IV. o balanceamento da carga;
- V. as restrições de carga e descarga manual (alcance dos braços do trabalhador);
- VI. os itens que devem ficar separados;
- VII. os itens que devem ficar agrupados; e
- VIII. as orientações dos itens (itens que não podem ser deitados ou virados de cabeça para baixo).

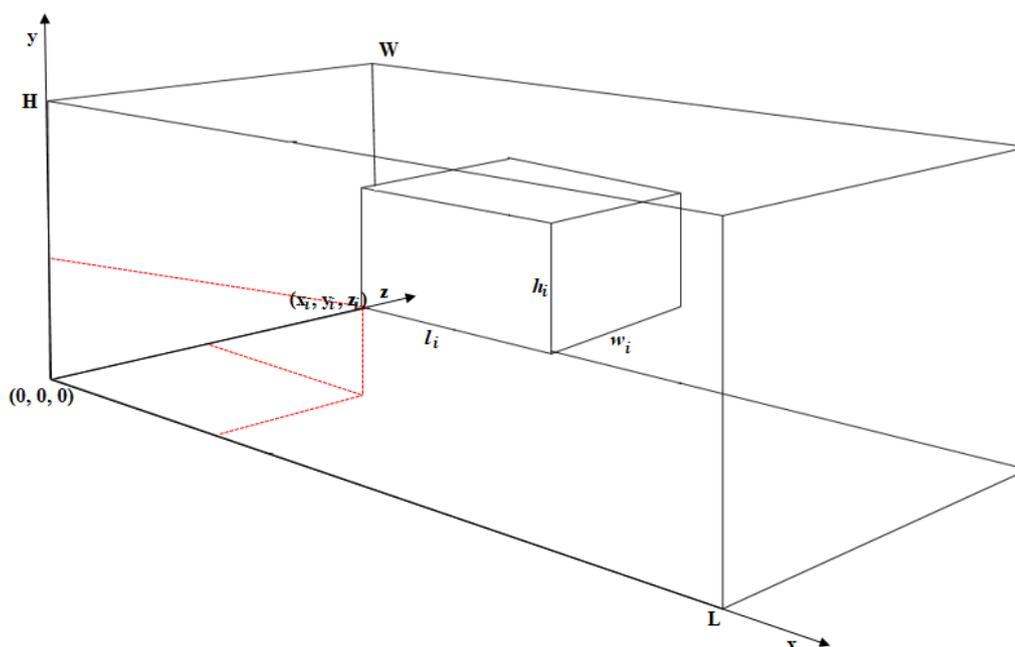
Contribuíram para o norteamento das resoluções dos PET os cálculos dos limites inferiores (*Lower Bounds*) para o valor da função objetivo. As metodologias para cálculo destes limites foram propostas por Martello et. al. (2000) e Boschetti (2004).

2.2 BASE PARA MODELAGEM DO PET

A base para modelagem do Problema do Empacotamento Tridimensional utilizada aqui fundamenta-se nos trabalhos de Chen et. al. (1995) e Mahvash et al. (2017). Para se modelar o PET parte-se do gráfico tridimensional com eixos x, y e z . Cada compartimento j está posicionado com seu canto inferior, de trás e esquerdo na origem dos eixos, ou seja, no ponto $(0,0,0)$. Os limites dos compartimentos nos eixos x, y e z são representados respectivamente pelos pontos L, W, H , que são parâmetros do modelo.

Os itens i têm suas dimensões representadas pelos parâmetros l_i, w_i, h_i que ficam, na orientação inicial do item, paralelos respectivamente aos eixos x, y e z . Quando acomodado em um compartimento j , a coordenada (x_i, y_i, z_i) corresponde às variáveis de decisão que representam a localização do vértice inferior, de trás e esquerdo do item i (Figura 2).

Figura 2 – Variáveis e Parâmetros PET



Fonte: Chen et. al. (1995)

Além disto, os itens podem assumir 6 (seis) orientações, representadas pelas variáveis binárias $(o_{1i}, o_{2i}, o_{3i}, o_{4i}, o_{5i}, o_{6i})$ onde 1 representa que o item está naquela orientação e 0, que o item não está naquela orientação. A Tabela 1

demonstra a qual eixo a dimensão do objeto é projetada em cada uma das seis orientações.

Outra representação necessária é da posição relativa entre os itens que são representados pelas variáveis binárias $a_{ii'}$, $b_{ii'}$, $c_{ii'}$, $d_{ii'}$, $e_{ii'}$ e $f_{ii'}$. Estas variáveis recebem o valor 1 quando o item i está respectivamente atrás, à frente, do lado esquerdo, do lado direito, abaixo e acima do item i' .

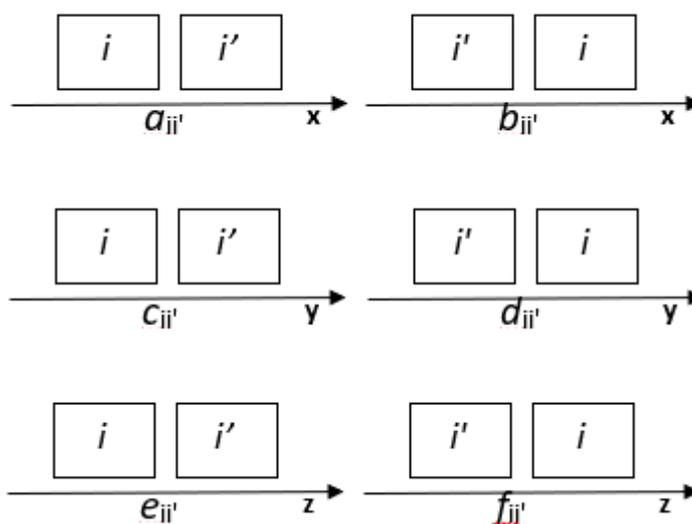
Tabela 1 – Orientações dos itens e projeções nos eixos

Orientação	x	y	z
o_1	l	w	h
o_2	w	l	h
o_3	h	w	l
o_4	w	h	l
o_5	l	h	w
o_6	h	l	w

Fonte: Mahvash et al., 2017.

Em relação ao gráfico, as variáveis $a_{ii'}$, $c_{ii'}$, e $e_{ii'}$ denotam que o item i está antes do item i' em relação ao eixo x, y e z , respectivamente. Já as variáveis $b_{ii'}$, $d_{ii'}$, e $f_{ii'}$ denotam que o item i' está antes do item i em relação ao eixo x, y e z , respectivamente (Figura 3).

Figura 3 - Posição relativa entre itens



Fonte: elaborada pelo autor

2.2 MODELAGEM DO PROBLEMA DE MAXIMIZAÇÃO DE SAÍDAS

O Problema de Maximização de Saídas é um PET que apresenta apenas um compartimento (KURPEL et. al., 2020). Pode ser conhecido também como o Problema de Carregamento de Contêiner ou o Problema da Mochila Tridimensional. Neste caso, dada uma quantidade de itens que excede a capacidade do compartimento, o objetivo é maximizar o volume carregado.

Pode-se chegar facilmente à modelagem do PET para um único compartimento fazendo adaptações na modelagem proposta por Chen et. al. (1995), conforme demonstrado por Silva et. al. (2019). As variáveis referentes a orientação dos objetos foram adaptadas do modelo proposto por Mahvash et al. (2017).

Os parâmetros e variáveis utilizados no modelo tem como base os princípios explicados anteriormente e estão descritos a seguir:

M	Um número arbitrariamente grande (<i>Big M</i>).
s_i	Variável binária que é igual a 1 se o item i está dentro do compartimento e 0 caso não.
l_i, w_i, h_i	Parâmetros indicando o comprimento, largura e altura do item i .
L, W, H	Parâmetros indicando o comprimento, largura e altura do contêiner j .
x_i, y_i, z_i	Variáveis inteiras (para localização) indicando a coordenada do canto da de trás, esquerdo, inferior do item i .
$o_{1i}, o_{2i}, o_{3i},$ o_{4i}, o_{5i}, o_{6i}	Variáveis binárias que indicam a orientação do item.
$a_{ii'}$	Variável binária que, se receber o valor 1, significa que o item i está atrás do item i' , caso contrário, recebe o valor 0.
$b_{ii'}$	Variável binária que, se receber o valor 1, significa que o item i está à frente do item i' , caso contrário, recebe o valor 0.
$c_{ii'}$	Variável binária que, se receber o valor 1, significa que o item i está no lado esquerdo do item i' , caso contrário, recebe o valor 0.
$d_{ii'}$	Variável binária que, se receber o valor 1, significa que o item i está no lado direito do item i' , caso contrário, recebe o valor 0.
$e_{ii'}$	Variável binária que, se receber o valor 1, significa que o item i está abaixo do item i' , caso contrário, recebe o valor 0.

$f_{ii'}$ Variável binária que, se receber o valor 1, significa que o item i está acima do item i' , caso contrário, recebe o valor 0.

O modelo matemático de programação inteira que representa o Problema do Empacotamento Tridimensional de Maximização de Saídas é o que se segue:

$$\text{Maximizar } \sum_{i \in I} l_i \cdot w_i \cdot h_i \cdot s_i$$

Sujeito a

$$x_i + l_i(o_{1i} + o_{5i}) + w_i(o_{2i} + o_{4i}) + h_i(o_{3i} + o_{6i}) \leq L + (1 - s_i) \cdot M \quad \forall i \quad (1)$$

$$y_i + l_i(o_{2i} + o_{6i}) + w_i(o_{1i} + o_{3i}) + h_i(o_{4i} + o_{5i}) \leq W + (1 - s_i) \cdot M \quad \forall i \quad (2)$$

$$z_i + l_i(o_{3i} + o_{4i}) + w_i(o_{5i} + o_{6i}) + h_i(o_{1i} + o_{2i}) \leq H + (1 - s_i) \cdot M \quad \forall i \quad (3)$$

$$x_i + l_i(o_{1i} + o_{5i}) + w_i(o_{2i} + o_{4i}) + h_i(o_{3i} + o_{6i}) \leq x_{i'} + (1 - a_{ii'}) \cdot M \quad \forall i, i' \quad i < i' \quad (4)$$

$$x_{i'} + l_{i'}(o_{1i'} + o_{5i'}) + w_{i'}(o_{2i'} + o_{4i'}) + h_{i'}(o_{3i'} + o_{6i'}) \leq x_i + (1 - b_{ii'}) \cdot M \quad \forall i, i' \quad i < i' \quad (5)$$

$$y_i + l_i(o_{2i} + o_{6i}) + w_i(o_{1i} + o_{3i}) + h_i(o_{4i} + o_{5i}) \leq y_{i'} + (1 - c_{ii'}) \cdot M \quad \forall i, i' \quad i < i' \quad (6)$$

$$y_{i'} + l_{i'}(o_{2i'} + o_{6i'}) + w_{i'}(o_{1i'} + o_{3i'}) + h_{i'}(o_{4i'} + o_{5i'}) \leq y_i + (1 - d_{ii'}) \cdot M \quad \forall i, i' \quad i < i' \quad (7)$$

$$z_i + l_i(o_{3i} + o_{4i}) + w_i(o_{5i} + o_{6i}) + h_i(o_{1i} + o_{2i}) \leq z_{i'} + (1 - e_{ii'}) \cdot M \quad \forall i, i' \quad i < i' \quad (8)$$

$$z_{i'} + l_{i'}(o_{3i'} + o_{4i'}) + w_{i'}(o_{5i'} + o_{6i'}) + h_{i'}(o_{1i'} + o_{2i'}) \leq z_i + (1 - f_{ii'}) \cdot M \quad \forall i, i' \quad i < i' \quad (9)$$

$$o_{1i} + o_{2i} + o_{3i} + o_{4i} + o_{5i} + o_{6i} = 1 \quad \forall i \quad (10)$$

$$a_{ii'} + b_{ii'} + c_{ii'} + d_{ii'} + e_{ii'} + f_{ii'} \geq s_i + s_{i'} - 1 \quad \forall i, i' \quad i < i' \quad (11)$$

$$a_{ii'}, b_{ii'}, c_{ii'}, d_{ii'}, e_{ii'}, f_{ii'}, s_i \in \{0,1\}$$

$$x_i, y_i, z_i \in Z^+$$

A função objetivo visa maximizar o volume carregado, enquanto as restrições de (1) a (3) garantem que os itens acomodados não ultrapassem as dimensões do compartimento. As restrições de (4) a (9) não permitem que os itens carregados se sobreponham entre si. Já a restrição (10) determina que cada item tenha uma e somente uma orientação. A restrição (11) é necessária para garantir que os itens carregados tenham uma posição relativa em relação uns aos outros.

Embora este seja um dos modelos de resolução do PET mais simples, ele apresenta grande sensibilidade à quantidade (n) de itens da instância a ser resolvida. A quantidade de variáveis é $10n + 3[(n - 1)^2 + n - 1]$, enquanto a quantidade de restrições é $4n + 3,5[(n - 1)^2 + n - 1]$.

Um método apresentado para diminuir o tempo de resolução do problema utilizando a modelagem matemática descrita anteriormente, consiste em descobrir

um conjunto menor de itens que poderia maximizar o volume carregado no compartimento. Para isto é feita uma relaxação das restrições de não sobreposição dos itens através do seguinte modelo de Programação Linear Inteira descrito por Nascimento et. al. (2021):

$$\text{Maximizar } \sum_{i \in I} v_i r_i$$

Sujeito a:

$$\sum_{i \in I} (l_i w_i h_i) r_i \leq LWH \quad \forall i \in I \quad (12)$$

$$r_i \geq r_{i+1} \quad \forall i \in I_k^+ \quad (13)$$

$$r_i \in \{0,1\}$$

A função objetivo visa maximizar o volume carregado, enquanto a restrição (12) garante que o volume carregado não é maior que o volume do contêiner. Este modelo não garante que a solução apresentada seja viável. Desta forma, deve-se verificar a viabilidade da solução, utilizando, por exemplo, o modelo para resolução do PET com um único compartimento adaptado de Chen et. al. (1995), descrito anteriormente. O problema é resolvido utilizando apenas os itens que fazem parte da solução do modelo relaxado. Caso a solução encontrada não seja viável, inclui-se esta solução ao conjunto I_k^+ e adiciona-se ao modelo relaxado a restrição de que este padrão não pode ser utilizado (13). Estes passos são repetidos até que seja encontrada uma solução viável, que, no caso, é uma solução ótima.

2.3 MODELAGEM DO PROBLEMA DE MINIMIZAÇÃO DE ENTRADAS

Chen et al. (1995) propôs um modelo matemático para resolução do problema que parte do pressuposto que existe uma quantidade limitada de compartimentos que podem ter dimensões diferentes, ou não, e existem uma quantidade finita de itens que devem ser acomodados nestes compartimentos. Este problema pertence ao grupo de minimização de entradas, entre os dois grupos de classificação de Problemas de Corte e Empacotamento (KURPEL et. al., 2020).

O modelo pode ser pensado adicionando a variável a seguir ao modelo do subitem 2.2.

n_j Variável binária que é igual a 1 se o compartimento j está sendo usado e 0 caso não.

Também é necessário incrementar mais uma dimensão as variáveis s_i que passa a ser s_{ij} , que denota se o item i é carregado no compartimento j . Se os compartimentos foram heterogêneos (com dimensões diferentes), também é necessário adicionar mais uma dimensão aos parâmetros L, W, H que passa a ser L_j, W_j, H_j .

Segue a modelagem baseada em Chen et al. (1995) e Mahvash et al. (2017) para o modelo de Minimização de Saídas com compartimentos homogêneos:

$$\text{Minimizar } \sum_m^{j=1} n_j$$

Sujeito a

$$x_i + l_i(o_{1i} + o_{5i}) + w_i(o_{2i} + o_{4i}) + h_i(o_{3i} + o_{6i}) \leq L + (1 - s_i).M \quad \forall i \quad (14)$$

$$y_i + l_i(o_{2i} + o_{6i}) + w_i(o_{1i} + o_{3i}) + h_i(o_{4i} + o_{5i}) \leq W + (1 - s_i).M \quad \forall i \quad (15)$$

$$z_i + l_i(o_{3i} + o_{4i}) + w_i(o_{5i} + o_{6i}) + h_i(o_{1i} + o_{2i}) \leq H + (1 - s_i).M \quad \forall i \quad (16)$$

$$x_i + l_i(o_{1i} + o_{5i}) + w_i(o_{2i} + o_{4i}) + h_i(o_{3i} + o_{6i}) \leq x_{i'} + (1 - a_{ii'}).M \quad \forall i, i' \quad i < i' \quad (17)$$

$$x_{i'} + l_{i'}(o_{1i'} + o_{5i'}) + w_{i'}(o_{2i'} + o_{4i'}) + h_{i'}(o_{3i'} + o_{6i'}) \leq x_i + (1 - b_{ii'}).M \quad \forall i, i' \quad i < i' \quad (18)$$

$$y_i + l_i(o_{2i} + o_{6i}) + w_i(o_{1i} + o_{3i}) + h_i(o_{4i} + o_{5i}) \leq y_{i'} + (1 - c_{ii'}).M \quad \forall i, i' \quad i < i' \quad (19)$$

$$y_{i'} + l_{i'}(o_{2i'} + o_{6i'}) + w_{i'}(o_{1i'} + o_{3i'}) + h_{i'}(o_{4i'} + o_{5i'}) \leq y_i + (1 - d_{ii'}).M \quad \forall i, i' \quad i < i' \quad (20)$$

$$z_i + l_i(o_{3i} + o_{4i}) + w_i(o_{5i} + o_{6i}) + h_i(o_{1i} + o_{2i}) \leq z_{i'} + (1 - e_{ii'}).M \quad \forall i, i' \quad i < i' \quad (21)$$

$$z_{i'} + l_{i'}(o_{3i'} + o_{4i'}) + w_{i'}(o_{5i'} + o_{6i'}) + h_{i'}(o_{1i'} + o_{2i'}) \leq z_i + (1 - f_{ii'}).M \quad \forall i, i' \quad i < i' \quad (22)$$

$$o_{1i} + o_{2i} + o_{3i} + o_{4i} + o_{5i} + o_{6i} = 1 \quad \forall i \quad (23)$$

$$a_{ii'} + b_{ii'} + c_{ii'} + d_{ii'} + e_{ii'} + f_{ii'} \geq s_{ij} + s_{i'j} - 1 \quad \forall i, i' \quad i < i', \forall j \quad (24)$$

$$\sum_m^{j=1} s_{ij} = 1 \quad \forall i \quad (25)$$

$$\sum_N^{i=1} s_{ij} \leq M \cdot n_j \quad \forall j \quad (26)$$

$$a_{ii'}, b_{ii'}, c_{ii'}, d_{ii'}, e_{ii'}, f_{ii'}, s_{ij}, n_j \in \{0,1\}$$

$$x_i, y_i, z_i \in Z^+$$

A função objetivo visa minimizar a quantidade de compartimentos utilizados. As restrições de (14) a (24) foram abordadas no subitem 2.2. A restrição (25) assegura que todos os itens sejam acomodados em um compartimento. A restrição (26) faz a ligação entre as restrições e a função objetivo, ao determinar que todo compartimento que tenha ao menos um item carregado, seja considerado utilizado.

Um dos desafios da resolução desta modelagem é determinar qual o número mínimo de compartimentos necessários para carregar todos os itens, já que, se não for determinado um número suficiente de compartimentos, o problema fica sem solução. Por outro lado, colocar como parâmetro um grande número excedente de compartimentos pode aumentar o tempo de resolução do problema, já que a quantidade de variáveis e restrições aumentam proporcionalmente à quantidade de compartimentos disponíveis.

Além da utilização de Programação Inteira, Martello et. al. (2000) desenvolveram um método exato, baseado em *Branch-and-Bound*, capaz de resolver PET sem rotação. O método conseguiu resolver algumas instâncias de até 60 itens na otimalidade com um limite de tempo de execução de 1000 segundos. O algoritmo foi codificado em Linguagem C e o experimento realizado em um HP9000/160 de 160 MHz.

2.5 MÉTODOS HEURÍSTICOS DE SOLUÇÃO

Segundo Fanslau e Bortfeldt (2010) existem 3 grupos de métodos heurísticos de resolução para os Problemas de Empacotamento Tridimensional:

- I. As heurísticas convencionais: Algoritmos Gulosos e de Busca Local;
- II. Meta-heurísticas: Algoritmos Genéticos, Busca Tabu, GRASP e Simulated Annealing; e
- III. Métodos de busca em árvore.

O método de busca em árvore foi utilizado por Fanslau e Bortfeldt (2010) e por Sheng et. al. (2014), ambos na resolução do problema de maximização de saída com um único compartimento.

Na resolução de problemas com um único compartimento com o objetivo de maximizar a quantidade de itens carregados, foram usados, entre outros métodos,

Algoritmo Genético (KANG et. al., 2012) e Simulated Annealing (EGEBLAD e PISINGER, 2009). O Algoritmo de Otimização da Colônia de Formigas foi utilizado por Levine e Ducatelle (2004) na resolução do Problema de Corte e Empacotamento Unidimensional e por Liang et. al. (2007) para resolução do problema de maximização de saída.

Nos Problemas de Empacotamento Tridimensional, a inserção de restrições práticas torna a solução ainda mais difícil. Alonso et. al. (2020) utilizou GRASP para resolução de problemas com restrições práticas, como por exemplo, peso total por compartimento e distribuição de peso por eixo de caminhões. Gzara et. al. (2020) apresentaram um método baseado em Geração de Colunas para a resolução de carregamento de pallets com restrições de estabilidade vertical. Este tipo de restrição exige “um percentual mínimo da base das caixas com apoio de outros itens ou do piso do contêiner” (KURPEL, 2015). O Algoritmo de Otimização da Colônia de Formigas também foi utilizado na resolução do Problemas de Empacotamento Tridimensional com múltiplos objetivos, em que, além de minimizar a quantidade de compartimentos utilizados, buscava-se também minimizar o Desvio-Padrão da distribuição do peso entre os compartimentos (LARA e LABRADOR, 2010). O segundo objetivo deste problema tem como resultado prático obter carregamentos mais balanceados, com a distribuição mais uniforme do peso entre os compartimentos utilizados.

A Busca Local Guiada foi utilizada para resolução do problema de minimização de entradas com compartimento homogêneo (FAROE et. al., 2003). Para o mesmo tipo de problema também foram utilizados GRASP/VND (PARREÑO et. al., 2008), Busca Tabu (CRAINIC et. al, 2009), Algoritmo Genético (FENG et. al., 2013), Algoritmo Genético com Chave Randômica (GONÇALVES e RESENDE, 2013) e Geração de Colunas (MAHVASH et. al., 2017).

Dos métodos citados anteriormente para a solução do problema de minimização de entradas sem restrições práticas e com um único objetivo, três não admitem a rotação de itens (FARAO et. al., 2003; PARREÑO et. al., 2008; CRAINIC et. al., 2009). Embora em aplicações práticas possa haver restrições aos 6 tipos de rotação possíveis, ao menos dois tipos de rotação são aceitáveis na maioria dos casos (FENG et. al., 2013).

Embora Feng et. al. (2013), Gonçalves e Resende (2013) e Mahvash et. al., (2017) tenham utilizado as mesmas instâncias, permitindo a comparação entre o

desempenho do método em relação a função objetivo, o mesmo não pode se dizer quanto ao desempenho do tempo computacional. Feng et. al. (2013) registraram ter limitado seu método baseado em Algoritmo Genético a 200 gerações e a 100 segundos para a resolução dos problemas sem rotação, mas não deixaram informações quanto aos limites na resolução dos problemas com rotação. Gonçalves e Resende (2013) informaram ter limitado o método, baseado em Algoritmo Genético com Chaves Randômicas, a apenas 200 gerações, sem limites de tempo. Mahvash et. al. (2017) impuseram um limite de 1000 segundos a resolução das instâncias e registram o tempo de resolução para cada classe de problema (Tabela 1).

A oportunidade de contribuição deste trabalho, além de gerar mais registros comparativos do tempo computacional para resolução do problema, está em procurar melhorar a eficiência do processo de solução, principalmente para as instâncias de 200 itens. Das 8 classes de problemas com 200 itens, o método baseado em Gerações de Colunas desenvolvido em Mahvash et. al. (2017) foi capaz de resolver apenas uma instância dentro do limite proposto de 1000 segundos.

3 METODOLOGIA

3.1 FORMULAÇÃO DO PROBLEMA

É importante mencionar que, para algumas instâncias, que a quantidade de itens é entre 69 e 469, nenhum desses modelos e métodos podem encontrar a solução. Esses resultados mostram que ainda existem espaço para melhoria dos métodos de carregamento de contêiner, especialmente se restrições práticas são levadas em consideração. (Nascimento et. al., 2021, p. 106)

O objetivo do problema utilizado neste trabalho é minimizar a quantidade de compartimentos utilizados, com as restrições de que todos os itens devem ser carregados ortogonalmente dentro dos limites físicos dos compartimentos, sem que haja sobreposição dos itens entre si. Para aplicação da resolução proposta, o problema deve conter compartimentos homogêneos e ilimitados e sem limitação de rotação dos itens.

3.2 MODELAGEM DO PROBLEMA

Foram experimentados quatro métodos de resolução do PET. A resolução pelo método de Geração de Colunas, desenvolvido por Mahvash et. al. (2017) e Estado da Arte, serviu como comparação para os outros três métodos criados no presente trabalho. O segundo método experimentado foi a utilização do Algoritmo de Otimização da Colônia de Formigas. O terceiro e o quarto método combinam o Algoritmo de Otimização da Colônia de Formigas e o Método de Geração de Colunas (ACO + CG). A diferença do terceiro para o quarto método é que o terceiro inicia a resolução pelo método ACO e refina a solução pelo GC, enquanto o quarto método adota a estratégia contrária (CG + ACO).

As meta-heurísticas implementadas, assim como alguns métodos de resolução do Problema do Empacotamento Tridimensional, tem como base a Heurística de Pontos Extremos, desenvolvida por Crainic et. al (2008). Para diversificar a ordem de carregamento dos itens na heurística construtiva, foi utilizado

o Algoritmo de Otimização da Colônia de Formigas e o Método de Geração de Colunas.

As decisões determinísticas do Algoritmo de Pontos Extremos são quanto a ordem de carregamento dos itens, rotação dos itens e escolha do Espaço Residual onde acomodar os itens. O ACO será utilizado para diversificar a ordem de carregamento dos itens, o que, indiretamente, contribui para a diversificação da escolha do Espaço Residual e da rotação dos itens.

3.2.1 Heurística de Pontos Extremos

Pontos Extremos é uma Heurística utilizada para a obtenção de uma solução viável para o Problema do Empacotamento Tridimensional com rotação. A noção de utilização de Pontos Extremos para obtenção de soluções para o PET foi introduzida por Crainic et. al (2008) como alternativa a utilização de Pontos de Canto. A Heurística pode ser utilizada para a construção de solução tanto para o problema bidimensional quanto para o problema tridimensional.

Algoritmo 1: Heurística de Pontos Extremos

entrada: SI, (L,W,H)

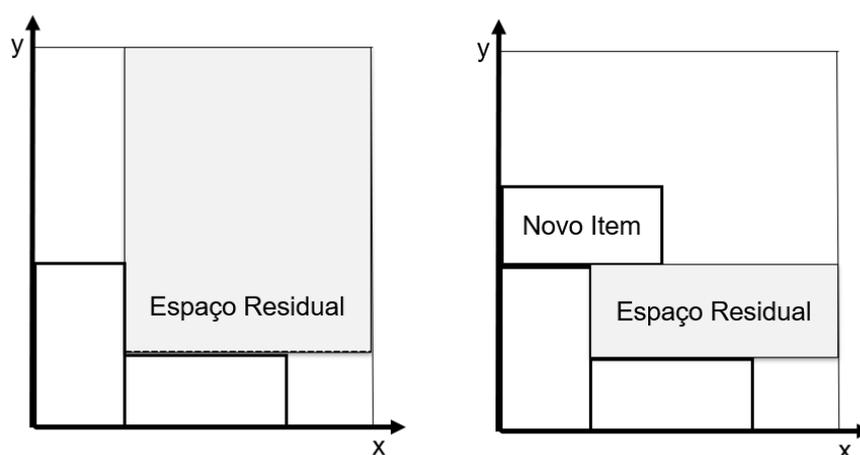
- 1 Inicializa um compartimento b com dimensões $L \times W \times H$
- 2 $PEs = \{(0,0,0)\}$
- 3 $ERs = \{(L,W,H)\}$
- 4 $PI = \emptyset$
- 5 **Para cada** i in SI **faça**
- 6 **Para cada** ERs **faça**
- 6 **se** ColocaltemNoCompartimento(i, ERs, PEs) **então**
- 7 Atualiza- $PEs(PI, i, ep, PEs)$
- 8 AtualizaEspaçoResidual(i, PEs)
- 9 Remove item i de SI e adiciona em PI
- 10 **Fim do se**
- 11 **Fim do para cada**
- 11 **se** não há espaço suficiente para acomodar o item **então**
- 12 **Inicializa outro compartimento e vai pro passo 1**
- 14 **Fim do para cada**

saída: PI

Para fins de exemplificação do Algoritmo vamos partir do pressuposto que os itens não são rotacionados, ou seja, sempre são carregados com a orientação o_{1i} . Para facilitar, vamos usar também as mesmas notações já abordadas no subitem 2.2. A Heurística de Pontos Extremos se inicia com um compartimento e um Ponto Extremo, situado na origem dos eixos x, y e z $(0, 0, 0)$. O Espaço Residual gerado é toda a área do compartimento.

Então é feita a colocação do primeiro item neste Ponto Extremo, que é excluído do grupo de PEs , juntamente com seu respectivo Espaço Residual. Após a inserção é necessário atualizar todos os Espaços Residuais em relação ao item inserido, pois este novo item pode aumentar a restrição de um Espaço Residual criado anteriormente.

Figura 4 – Atualização do Espaço Residual



Fonte: Crainic et. al., 2008.

A partir do novo item inserido também são gerados três novos Pontos Extremos temporários, conforme as fórmulas a seguir:

$$p_i^1 = (x_i + l_i, y_i, z_i) \quad (27)$$

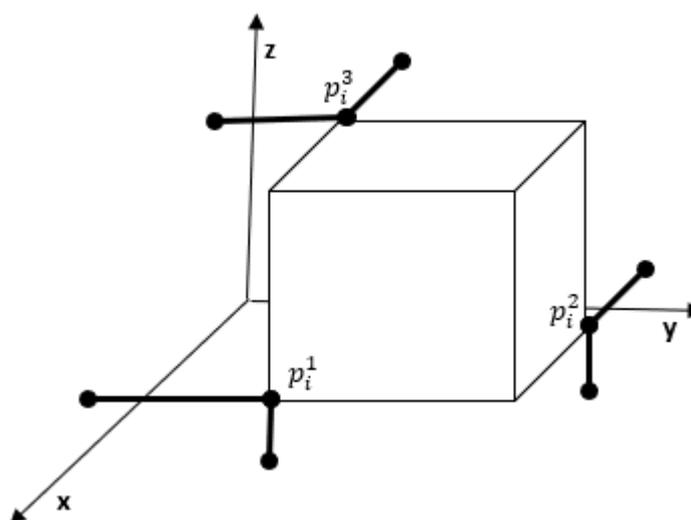
$$p_i^2 = (x_i, y_i + w_i, z_i) \quad (28)$$

$$p_i^3 = (x_i, y_i, z_i + h_i) \quad (29)$$

Cada Ponto Extremo temporário pode ser decomposto em dois novos Pontos Extremos, realizando a projeção nas paredes do compartimento e nos itens

já inseridos no compartimento. O Ponto Extremo temporário p_i^1 é projetado na parede esquerda do compartimento e nos itens à esquerda do ponto. O mesmo ponto também é projetado na parede de baixo do compartimento e nos itens abaixo do ponto inseridos no compartimento. O Ponto Extremo temporário p_i^2 é projetado na parede do fundo do compartimento e nos itens que estão atrás do ponto. O mesmo ponto também é projetado na parede de baixo do compartimento e nos itens abaixo do ponto. Já o Ponto Extremo temporário p_i^3 é projetado na parede do fundo do compartimento e nos itens que estão atrás do ponto e na parede esquerda do compartimento e nos itens à esquerda do ponto. Em outras palavras, p_i^1 se projeta paralelo eixos y e z , p_i^2 se projeta paralelo aos eixos x e z e p_i^3 se projeta paralelo aos eixos x e y (Figura 5).

Figura 5 – Pontos Externos criados por um item



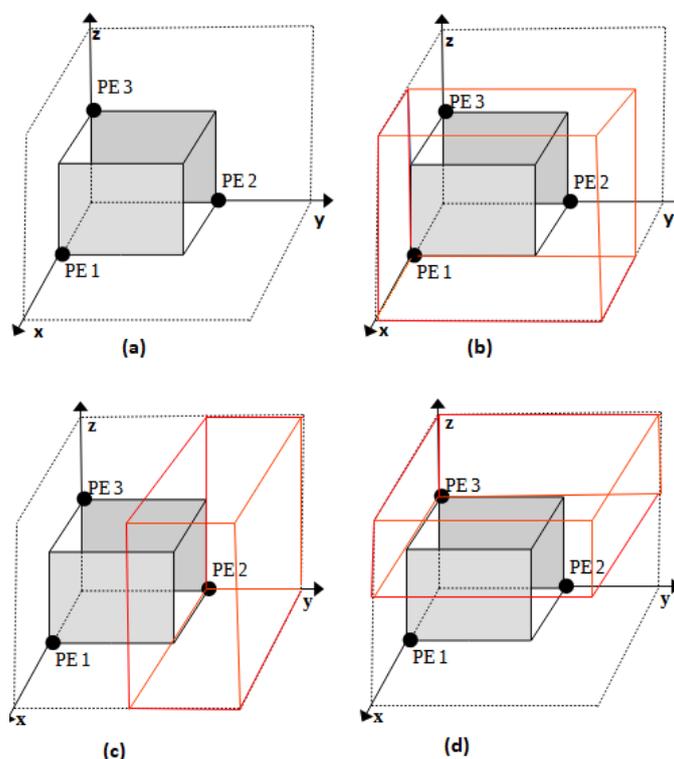
Fonte: Crainic et. al., 2008.

Cada Ponto Extremo projetado é atualizado a cada projeção se o valor da coordenada do ponto no eixo paralelo à projeção for maior que o valor da mesma coordenada na última projeção. Ou seja, se um ponto apresentar duas projeções (uma na parede do compartimento e outra em um item), o ponto que será mantido será o que estiver mais próximo do Ponto Extremo temporário que gerou a projeção. Neste caso será o Ponto Extremo projetado no item, já que o item está entre a parede do compartimento e o ponto temporário. Com essas atualizações, ao final cada Ponto Extremo temporário irá gerar apenas dois novos Pontos Extremos.

Para cada um dos seis novos Pontos Extremos é verificado se já não existe outro igual no conjunto de PEs. Se não existir, esse novo Ponto Extremo é adicionado ao conjunto de PEs.

Cada Ponto Extremo adicionado ao PEs gerará um novo Espaço Residual. O espaço residual será projetado na direção oposta à origem dos eixos até o limite do objeto mais próximo do Ponto Extremo que o gerou, seja um outro item ou as paredes do compartimento. Na Figura 6, os Espaços Residuais são representados pelos paralelepípedos retângulos vermelhos.

Figura 6 – Exemplo de Espaços Residuais



Fonte: elaborada pelo autor

Os procedimentos descritos são realizados a cada inserção de novo item no compartimento. Quando não for possível adicionar um item a nenhum Espaço Residual existente, cria-se outro compartimento com Ponto Extremo $(0,0,0)$ e seu respectivo Espaço Residual.

Um ponto importante da Heurística consiste em classificar os itens a serem carregados por algum critério. Também há critérios definidos quanto à escolha do Espaço Residual a ser utilizado e qual a orientação de cada item. É difícil gerar um

meio de diversificação dos dois primeiros fatores, já que existe uma grande combinação de orientações e escolha de Espaços Residuais, quanto combinados. Alterações na orientação geram alterações na criação dos Espaços Residuais e alteração da Escolha dos Espaços Residuais geram alterações na criação de novos Espaços Residuais e nas possibilidades de orientação dos outros itens.

A escolha do Espaço Residual no presente trabalho foi realizada priorizando os ER's com menor volume. Se o ER apresentar volume menor que o volume do item a ser inserido, é verificado se há possibilidade de inserção deste item neste Espaço Residual. A orientação do item seguiu as seguintes prioridades:

- i. Maior lado do item paralelo ao eixo y e segundo maior lado paralelo ao eixo x;
- ii. Maior lado do item paralelo ao eixo y e segundo maior lado paralelo ao eixo z;
- iii. Maior lado do item paralelo ao eixo x e segundo maior lado paralelo ao eixo y;
- iv. Maior lado do item paralelo ao eixo x e segundo maior lado paralelo ao eixo z;
- v. Maior lado do item paralelo ao eixo z e segundo maior lado paralelo ao eixo y; ou
- vi. Maior lado do item paralelo ao eixo z e segundo maior lado paralelo ao eixo x.

Seguindo a ordem de classificação dos Espaços Residuais, tenta-se inserir o item, obedecendo a prioridade de orientação. Caso não seja possível, passa-se para o próximo Espaço Residual.

A diversificação para melhorar o resultado da Heurística consiste em alterar a ordem de carregamento dos itens e isto pode ser feito através do Algoritmo de Otimização da Colônia de Formigas e do Método de Geração de Colunas.

3.2.2 O Algoritmo de Otimização da Colônia de Formigas

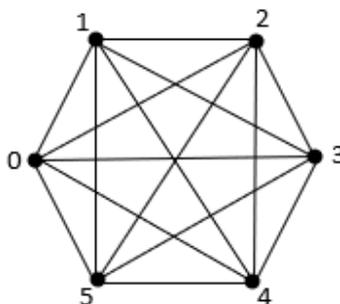
O Algoritmo de Otimização da Colônia de Formigas é uma meta-heurística introduzida por M. Dorigo para a solução de problemas de otimização combinatória. Trata-se de um algoritmo utilizado para a obtenção de boas soluções, quando não

se consegue a solução ótima em um tempo computacional viável (DORIGO e BLUM, 2005). A inspiração do algoritmo foi no comportamento de formigas reais na busca por comida. Inicialmente elas exploram o ambiente de forma aleatória, após encontrar comida, deixam um rastro de feromônio que depende da qualidade da comida. A comunicação das formigas através do feromônio permite que elas encontrem o caminho mais curto para a comida.

O ACO usa um processo iterativo que constrói uma solução a partir da trilha de feromônios e atualiza o depósito de feromônios nas trilhas baseado nas melhores soluções encontradas.

No caso do Problema do Empacotamento Tridimensional, pode se considerar que cada item é representado por um vértice i (Figura 7). Iniciando-se do vértice 0 (que não representa um item, apenas o ponto inicial) deve se passar uma vez por todos os itens percorrendo as arestas ii' . A sequência de carregamento dos itens é determinada pelo caminho percorrido.

Figura 7 - Representação do caminho a ser percorrido pelas formigas



Fonte: elaborada pelo autor

Um dos parâmetros que deve ser determinado no ACO é a quantidade de formigas que percorrerão os caminhos a cada iteração. Neste trabalho, baseado em Duan et. al. (2007), a quantidade de formigas (A) foi determinada em razão (Q) da quantidade de itens (n) da instância a ser resolvida, conforme a fórmula a seguir:

$$A = Q \cdot n \quad (30)$$

O Algoritmo de Otimização da Colônia de Formigas se inicia atribuindo um valor de feromônio $\tau_i^{i'}$ a cada aresta ii' . Também é atribuído um valor heurístico $c_i^{i'}$ a cada aresta, que, no caso, foi o valor do volume do item i' . Em seguida, a cada iteração é calculada a função $p_i^{i'}$, que descreve a probabilidade de, partindo do vértice i , ir para o vértice i' , ou seja, de se carregar o item i' após o carregamento do item i . Esta função é dada por:

$$p_i^{i'} = \frac{[\tau_i^{i'}]^\alpha \cdot [\eta(c_i^{i'})]^\beta}{\sum_{c_i^{i'} \in R} [\tau_i^{i'}]^\alpha \cdot [\eta(c_i^{i'})]^\beta} \quad \forall i, i' \quad i \neq i' \quad (31)$$

O parâmetro α controla a importância dos feromônios, enquanto o parâmetro β representa a importância da heurística. O parâmetro η aumenta ou diminui o valor da heurística. No presente trabalho ele foi calculado conforme a seguir:

$$\eta = Q_\eta \cdot \max(l_i \cdot w_i \cdot h_i) \quad (32)$$

Onde η é a dado pelo volume máximo que os itens podem assumir, representado por $\max(l_i \cdot w_i \cdot h_i)$, multiplicado por Q_η , que é um parâmetro ajustado mediante teste.

Para a diversificação da ordem de carregamento de itens, para cada formiga, a probabilidade $p_i^{i'}$ é multiplicada por um número randômico entre 0 e 1, representado por $r_i^{i'}$ na fórmula (33).

$$o_i^{i'} = p_i^{i'} \cdot r_i^{i'} \quad \forall i, i' \quad i \neq i' \quad (33)$$

Este procedimento proporciona que sequências com maior probabilidade tenham mais chances de serem utilizadas, mas não descarta o uso de sequências com baixas probabilidades. O maior $o_i^{i'}$ determina qual caminho será percorrido a partir de um vértice, ou seja, qual item i' será carregado após o item i .

Após montado o caminho, que representa a sequência de carregamento, esta informação é utilizada Heurística de Pontos Extremos (abordada no Subitem 3.2.1) para gerar uma solução.

Para os valores da função objetivo foi utilizada uma estratégia de tornar a solução inteira em uma solução racional. Isto foi realizado somando, ao valor da função objetivo, o menor índice de utilização do compartimento (u_j) encontrado na respectiva solução. O índice de utilização do compartimento é obtido através da soma do volume dos itens inseridos em no compartimento j dividido pelo volume do compartimento. A variável binária a_i^j revela se o item i está no compartimento j ou não. Segue as formulações:

$$u_j = \frac{\sum_{i=1}^n a_i^j \cdot l_i \cdot w_i \cdot h_i}{L \cdot W \cdot H} \quad (34)$$

$$\xi = \text{Obj} + \min(u_j) \quad (35)$$

Montada uma solução (ξ) para cada formiga, é verificado, ao final da iteração, qual foi a melhor solução (S_{Iter}). A partir desta solução é realizada a atualização da trilha de feromônio, conforme a fórmula a seguir:

$$\tau_{i'}^{i'} \leftarrow (1 - e) \cdot \tau_{i'}^{i'} + \Delta\tau_{i'}^{i'} \cdot \left(p + \frac{L}{S_{Iter}} \cdot a \right) \quad \forall i, i' \quad i \neq i' \quad (36)$$

Fonte: adaptado de Dorigo e Gambardella, 1997.

Onde $\tau_{i'}^{i'}$ é o valor do feromônio depositado na aresta ii' e $(1 - e) \cdot \tau_{i'}^{i'}$ é a evaporação do feromônio na proporção do parâmetro e . A variável binária $\Delta\tau_{i'}^{i'}$ assume o valor 1 se o a aresta ii' foi percorrida na melhor solução da iteração, e assume o valor 0, caso contrário. O parâmetro p representa a quantidade de feromônio depositado. Baseado na Estrutura do Hipercubo, proposta por Dorigo e Blum (2005), soma-se ao feromônio depositado um acréscimo a em razão da qualidade da solução da iteração $\frac{L}{S_{Iter}}$. Esta qualidade é medida pelo *Lower Bound* (L) dividido pelo valor da função objetivo da melhor solução da iteração (S_{Iter}). O L é o *Lower Bound* calculado a partir da soma entre o volume de todos os itens dividido pelo volume do compartimento, conforme a seguir:

$$L = \frac{\sum_{i=1}^n l_i \cdot w_i \cdot h_i}{L \cdot W \cdot H} \quad (37)$$

Baseado em Martello et. al. (2000)

Ao final de cada iteração, além de atualizar os valores dos feromônios, é verificado se o valor da melhor solução da iteração é menor que o valor da melhor Solução Geral (S_{bs}), caso positivo, atualiza-se a melhor Solução Geral.

O algoritmo continua retornando ao cálculo da probabilidade (Fórmula 31), até que se atinja o critério de parada.

Algoritmo 2: Estrutura do Algoritmo de Colônia de Formigas

entrada: $\mathcal{P} = (SI, (L,W,H))$

```

1  InicializaValoresFeromônio( $\mathcal{T}$ )
2   $\xi_{bs} \leftarrow$  Nulo
3  Enquanto não chegar no critério de parada faça
4       $\xi_{iter} \leftarrow \emptyset$ 
5       $p \leftarrow$  CalculaProbabilidade( $\mathcal{T}$ )
6      Para  $j=1, \dots, n$  faça
7           $o \leftarrow$  Ordenaltens( $p$ )
8           $\xi \leftarrow$  ExtremPoints( $o, SI, L, W, H$ )
9              se ( $f(\xi) < f(\xi_{iter})$ ) ou ( $\xi_{iter} =$  Nulo ) então  $\xi_{iter} \leftarrow \xi$ 
10     Fim do para
11     AplicaAtualizaçãoFeromônio( $\mathcal{T}, \xi_{iter}$ )
12     se ( $f(\xi_{iter}) < f(\xi_{bs})$ ) ou ( $\xi_{bs} =$  Nulo ) então  $\xi_{bs} \leftarrow \xi_{iter}$ 
13 Fim do enquanto
saída: Melhor solução  $\xi_{bs}$ 

```

Fonte: Dorigo e Blum, 2005.

Desta forma, o Algoritmo de Otimização das Colônias de Formigas é utilizado como estratégia de diversificação e intensificação da exploração do espaço solução.

3.2.2.1 Simplificando o ACO

A forma apresentada anteriormente de aplicação do Algoritmo da Otimização da Colônia de Formigas ao Problema do Empacotamento Tridimensional com rotação tem a desvantagem na relação entre a quantidade de itens n e a quantidade de arestas geradas. De fato, a quantidade de arestas é igual a $n!$ o que pode diminuir a eficiência do algoritmo na resolução de problemas com grandes instâncias.

Visando diminuir o número de variáveis, foi alterada a forma de depósito de feromônio na trilha no procedimento `AplicaAtualizaçãoFeromônio`, utilizando a variável k_i que representa a ordem de carregamento do item i , multiplicado por d , que é um parâmetro que representa o fator de decréscimo do feromônio, conforme a fórmula a seguir:

$$\tau_i \leftarrow (1 - e)\tau_i + p - (k_i - 1) \cdot d + \frac{L}{S_{Iter}} \cdot a \quad \forall i \quad (36')$$

Assim, cada item é marcado com uma quantidade de feromônio p que é diminuída por um fator de decréscimo d . Este fator de decréscimo aumenta de acordo com a ordem de carregamento do item k_i , fazendo com que um item carregado posteriormente seja marcado com menos feromônio que um item carregado anteriormente. Já na rotina `SoluçãoConstrutiva`, a fórmula aplicada para obtenção da probabilidade foi adaptada, conforme a seguir:

$$p_i = \frac{[\tau_i]^\alpha \cdot [\eta(c_i)]^\beta}{\sum_{c_i \in I} [\tau_i]^\alpha \cdot [\eta(c_i)]^\beta} \quad \forall i \quad (31')$$

Neste caso, p passa a representar a probabilidade de prioridade de carregamento. Finalmente, o carregamento é realizado pela classificação decrescente dos itens pelo o_i , determinado pela fórmula abaixo, adaptada da fórmula (33):

$$o_i = p_i \cdot r_i \quad \forall i \quad (33')$$

Através destas modificações, é possível melhorar a relação entre o número de itens n e a probabilidade de prioridade de carregamento, passando a quantidade de variáveis o_i ser igual a n , otimizando-se assim o desempenho do algoritmo.

3.2.2.2 Parâmetros

O Algoritmo de Otimização da Colônia de formigas possui alguns parâmetros que ajustam seu funcionamento. Neste trabalho foram utilizados os parâmetros α (importância do feromônio), β (importância da heurística), Q_η (proporção do valor da heurística), Q (proporção de formigas), p (feromônio excretado), e (índice de evaporação), a (índice de acréscimo em razão da qualidade da solução), e d (valor do decréscimo em relação a ordem de carregamento do item).

Conforme Duan et. al. (2007) o desafio do ACO é atingir o balanceamento entre intensificação e exploração. A intensificação é baseada em experiências anteriores e exploração é a busca de novas descobertas. Este ponto de equilíbrio é buscado através da modulação dos parâmetros.

Ainda segundo Duan et al. (2007), os parâmetros influenciam a convergência, que é a melhora da solução ao longo do tempo. A convergência tem dois aspectos: velocidade e qualidade. Grandes velocidades de convergência podem prender a solução em um ótimo local, diminuindo sua qualidade, enquanto a qualidade da convergência pode penalizar sua velocidade.

A Tabela 2 apresenta a relação entre os parâmetros e a qualidade e velocidade da convergência. Relação direta significa que aumentar o valor do parâmetro aumenta o aspecto da convergência e a relação inversa diminui o aspecto da convergência ao aumentar o valor do parâmetro.

Tabela 2 - Relação entre os parâmetros do ACO e a convergência da solução

Parâmetro	Convergência	
	Qualidade	Velocidade
e (índice de evaporação)	Direta	Inversa
Q (proporção de formigas)	Direta	Inversa
α (importância do feromônio)	Direta	Inversa
β (importância da heurística)	Inversa	Direta
p (feromônio excretado)	Inversa	Direta

Fonte: Duan et al., 2007.

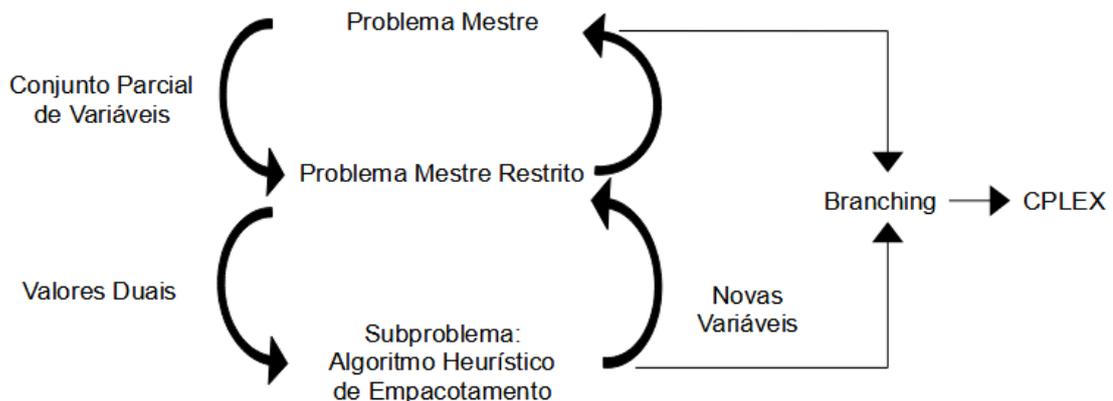
3.2.3 Método de Geração de Colunas

Mahvash et al. (2017) utilizou um método baseado em Geração de Colunas para resolução do Problema do Empacotamento Tridimensional com rotação. O método de Geração de Colunas foi proposto por Gilmore e Gomory (1963) e

utilizado inicialmente para a resolução do problema do corte de estoque unidimensional.

O Método de Geração de Colunas parte do princípio do particionamento do Problema de Programação Inteira e da utilização da relaxação de um Problema de Programação Linear para se conseguir uma boa aproximação de uma solução viável (Barnhart et. al., 1998). O CG parte da relaxação da Decomposição de Dantzig-Wolf, no que se denomina Problema Mestre. Do conjunto parcial de variáveis pertencentes ao Problema Mestre, é criado o Problema Mestre Restrito. A última parte do CG é o Subproblema, que tem a função de gerar novas variáveis (colunas) para o Problema Mestre Restrito. A estrutura do método na resolução do Problema do Empacotamento Tridimensional está representada na Figura 8.

Figura 8 - Iterações entre diferentes partes da Técnica de Geração de Colunas



Fonte: Mahvash et al., 2017.

Para o PET, o Problema Mestre é definido como o seguinte problema de particionamento (Ogunranti e Oluleye, 2016):

$$\text{mimimizar } \sum_{j=1}^n X_j$$

sujeito a:

$$\sum_{j=1}^n a_i^j \cdot X_j = 1 \quad \forall i \quad (38)$$

$$X_j \geq 0$$

A variável de decisão X_j na Decomposição de Dantzig-Wolf pode ser binária ou inteira e representa se o padrão j será utilizado (1 para utilizado e 0 para não utilizado) ou o quanto daquele padrão será utilizado, no caso de variável inteira. Por padrão, ou coluna, entende-se uma forma viável de se se carregar itens em um compartimento, no caso do Problema do Empacotamento Tridimensional. Esta variável é relaxada, transformando em um modelo de Otimização Linear, chamado Problema Mestre.

O parâmetro a_i^j representa se o item i está no compartimento j (1 caso esteja, 0 caso contrário). Desta forma, a função objetivo buscar minimizar a quantidade de compartimentos utilizados (padrões), com a restrição que todos os itens devem ser carregados.

Como pode ser custoso a enumeração e todas as variáveis do problema e a resolução do Problema Mestre pode ser computacionalmente inviável, utiliza-se um subconjunto de variáveis do Problema Mestre, denominado Problema Mestre Restrito (LÜBBECKE e DESROSIERS, 2005). A iniciação do Problema Mestre Restrito é feita com uma quantidade pequena de padrões que são obtidos, no Problema do Empacotamento Tridimensional, através da Heurística de Pontos Extremos. A partir das variáveis duais dos itens obtidas pela resolução do Problema Mestre Restrito, é construído um Subproblema. No exemplo do corte de estoque unidimensional, a modelagem do Subproblema corresponde ao problema da mochila:

$$\text{maximizar } \sum_{i=1}^n \pi_i \cdot x_i$$

sujeito a:

$$\sum_{j=1}^n w_j \cdot x_j \leq W \quad \forall i \quad (39)$$

$$x_i \in Z^+$$

A variável x_i representa se o item i será carregado ou não (1 se for carregado, 0 caso não). O parâmetro π_i é o valor da variável dual do item i , o parâmetro w_i é o tamanho do item i e o parâmetro W é a capacidade total da

mochila. Neste caso, o objetivo do Subproblema é maximizar o valor dos itens carregados, dada a restrição que todos os itens carregados não devem ultrapassar a capacidade da mochila. No problema de corte de estoque unidimensional, a mochila é a peça em estoque e os itens são os cortes demandados. No caso do Problema do Empacotamento Tridimensional, a mochila representa um compartimento, por isso pode-se chamar o Subproblema de Problema da Mochila Tridimensional. A modelagem do subproblema para o Problema do Empacotamento Tridimensional é o modelo apresentado na Subseção 2.2, substituindo o parâmetro l_i, w_i, h_i da função objetivo por π_i .

A partir da resolução do Subproblema, o novo padrão apenas é inserido no Problema Mestre Restrito caso o padrão tenha o custo reduzido (c_j) negativo. O custo reduzido é calculado a partir do formulação a seguir, baseada em Mahvash et. al.(2017):

$$c_j^\pi = 1 - \sum_{i=1}^n a_i^j \pi_i \quad (40)$$

Conforme mencionado anteriormente, a modelagem do Suproblema para o PET é o Problema de Maximização de Entradas, que também é *NP*-Difícil. Como alternativa, Mahvash et. al (2017) utilizou a Heurística de Pontos Extremos com os itens ordenados conforme os critérios a seguir.

- i. Ordem decrescente de π_i ;
- ii. Ordem decrescente de $\frac{\pi_i}{l_i \cdot w_i \cdot h_i}$;
- iii. Ordem decrescente de $\frac{\pi_i}{l_i + w_i + h_i}$;
- iv. Ordem decrescente de $\frac{\pi_i}{l_i}$;
- v. Ordem decrescente de $\frac{\pi_i}{w_i}$; e
- vi. Ordem decrescente de $\frac{\pi_i}{h_i}$.

A cada iteração são resolvidos 6 Subproblemas, utilizando a Heurística de Pontos Extremos, cada um com um tipo de ordenamento de itens. Todas os novos padrões com custo reduzido negativo gerados são inseridos no Problema Mestre Restrito.

As iterações continuam com a resolução do Problema Mestre Restrito, agora com as novas colunas inseridas (padrões), que gerará novos valores de variáveis duais para os itens, que modificará o Subproblema que, por sua vez, após resolvido, poderá gerar novas colunas para o Problema Mestre Restrito. O algoritmo é interrompido quando não se gerar ao menos um padrão com o custo reduzido negativo ou quando se atingir um critério de parada.

Se a solução encontrada no Problema Mestre Restrito for inteira, esta solução é a solução do problema. Caso contrário, é necessário alterar o problema, modificando as variáveis para inteiras.

Ao modificar as variáveis para inteiras, existe a possibilidade do Problema Mestre Restrito não ter solução viável (Barnhart et. al., 1998). Um dos modos de se evitar esta impossibilidade é alterar condição de igualdade da restrição, conforme a seguir:

$$\sum_{j=1}^n a_i^j \cdot X_j \geq 1 \quad \forall i \quad (39')$$

Tal relaxação permite que a solução obtida tenha um ou mais itens destinados a mais de um compartimento. Esta inconsistência é facilmente contornada excluindo da solução os itens repetidos.

Algoritmo 3: Estrutura do Algoritmo de Colônia de Formigas

entrada: $\mathcal{P} = (SI, (L,W,H))$

- 1 Gera os padrões iniciais $X \leftarrow \text{ExtremPoints}(SI, L, W, H)$
- 2 **Enquanto** não chegar no critério de parada **faça**
- 3 Resolve o Problema Mestre Restrito
- 4 Resolve o Subproblema utilizando os valores duais da resolução do Problema Mestre Restrito
- 5 **Para** cada novo Padrão x gerado pela resolução do Subproblema **faça**
- 6 **se** custo reduzido de negativo, $X \leftarrow x$
- 7 **Fim do para**
- 8 **Fim do enquanto**
- 9 **Se** solução do Problema Mestre Restrito não é inteira
- 10 Altera Problema Mestre Restrito para Problema de Otimização Inteira
- 11 Altera sinal da Restrição do Problema Mestre Restrito
- 12 Soluciona Problema Mestre Restrito
- 13 **Fim do Se**

saída: Solução do Problema Mestre Restrito

Fonte: baseado em Mahvash et al., 2017.

3.2.4 Algoritmo de Otimização da Colônia de Formigas e Método de Geração de Colunas

A utilização dos dois métodos, iniciando pelo Colônia de Formigas, parte do pressuposto que os padrões gerados nas diversas iterações do Algoritmo de Otimização da Colônia de Formigas podem ser combinados e se chegar a uma solução melhor.

Primeiro são realizadas as iterações do ACO, após, todos os compartimentos gerados nessas iterações são utilizados como padrões. Esses padrões são inseridos no Problema Master, com a modificação da restrição para evitar que o problema não tenha solução (39'). A partir da resolução do Problema Master, é realizada a exclusão dos itens que porventura estiverem repetidos nos compartimentos.

3.2.5 Método de Geração de Colunas e Algoritmo de Otimização da Colônia de Formigas

A utilização dos dois métodos, iniciando pelo Método de Geração de Colunas, parte do pressuposto que quanto mais tempo se executa o Método de Geração de Colunas, maior o esforço para se obter um incremento na função objetivo da solução final. Deste pressuposto vem a hipótese de que pode-se alcançar este mesmo incremento, com menor esforço, utilizando uma estratégia de intensificação. Esta estratégia de intensificação é obtida através do Algoritmo de Otimização da Colônia de Formigas.

O Método de Geração de Colunas é aplicado em sua totalidade ao problema, até a resolução do Problema Mestre. O compartimento que tiver o menor aproveitamento de espaço é desfeito utilizando a Fórmula (34). Nesta nova etapa, obtém-se uma Solução Parcial que é uma unidade a menos que a solução da Geração de Colunas e alguns itens que não estão carregados. Faroe et al. (2003) utilizaram esta estratégia para realização de Busca Local. No presente trabalho foi utilizado o Algoritmo de Otimização da Colônia de Formigas para consolidar a Solução Parcial.

Os itens da instância do Problema, nesta etapa, são divididos em dois tipos de grupos: F^j e A . Os grupos F^j são compostos pelos itens que foram carregados no compartimento j pelo Método de Geração de Colunas. O grupo A é composto pelos itens que foram retirados do compartimento.

Os itens do grupo A são atribuídos aos compartimentos j , utilizando-se o Algoritmo de Otimização da Colônia de Formigas. Os valores de feromônios iniciais (τ_i^j) da atribuição do item i a um compartimento j são inicialmente calculados como se segue:

$$\tau_i^j = p - (Q^j - s_i^j + 1) \cdot d \quad \forall i \in F^j \quad (41)$$

$$\tau_i^j = d \quad \forall i \in A, \forall j \quad (42)$$

O parâmetro Q^j é a quantidade de itens carregados no compartimento j e s_i^j é a ordem de carregamento do item na solução pelo Método de Geração de Colunas. Os parâmetros p (feromônio excretado) e d (índice de decréscimo), segue o descrito no Subitem 3.2.2.

A probabilidade de um item i ser atribuído a um compartimento j é dado por:

$$p_i^j = \frac{[\tau_i^j]^\alpha \cdot [c_i^j]^\beta}{\sum_{p_i^j \in P} [\tau_i^j]^\alpha \cdot [c_i^j]^\beta} \quad \forall i \in A, \forall j \quad (43)$$

O valor da heurística c_i^j é obtido através do cálculo (41), que é o volume do item dividido pelo volume ocioso do compartimento j :

$$c_i^j = \frac{l_i \cdot w_i \cdot h_i}{L \cdot W \cdot H - \sum_{n \in F^j} a_n^j \cdot l_n \cdot w_n \cdot h_n} \quad \forall i \in A, \forall j \quad (44)$$

Com base no cálculo de probabilidade é obtido a classificação das prioridades de designação dos itens, multiplicando o percentual de probabilidade por uma variável randômica (32). Estas prioridades são ordenadas de forma decrescente, então são realizadas as designações do maior para o menor valor, obedecendo aos seguintes princípios:

- i. cada item só é atribuído a um compartimento;
- ii. para se atribuir um item a um compartimento, o volume ocioso do compartimento deve ser maior ou igual ao volume do item;
- iii. quando um item for atribuído a um compartimento, o volume ocioso do compartimento é atualizado; e
- iv. é necessário atribuir todos os itens a um compartimento.

Após realizadas as atribuições, para cada compartimento que tiver itens novos atribuídos, será resolvido um problema com o intuito de carregar todos os itens. A ordem de carregamento em cada problema é obtida através da aplicação do Algoritmo de Otimização da Colônia de Formigas, conforme o Subitem 3.2.2. Vale destacar que cada item pertencente ao conjunto A possui um valor de feromônio para classificação da ordem de carregamento deste item em cada compartimento (τ_i^j). Os itens pertencentes aos grupos F^j só possuem um valor de feromônio para classificar a ordem de carregamento do item, pois os mesmos têm a atribuição fixada.

A atualização do feromônio de ordenamento do carregamento dos itens também sofre adaptação na parte do acréscimo de feromônio, descrita a seguir.

$$\tau_i^j \leftarrow (1 - e) \cdot \tau_i^j + p + \frac{a}{v} \quad \forall i \in A \quad (45)$$

$$\tau_i \leftarrow (1 - e) \cdot \tau_i + p + \frac{a}{v} \quad \forall i \in F^j \quad (46)$$

O princípio é mesmo das atualizações de feromônios apresentadas anteriormente. Somente o critério de acréscimo é alterado, dividindo o índice de acréscimo (a) pelo volume dos itens do conjunto A que não foram carregados na solução (v). Só são atualizados os feromônios τ_i^j e τ_i se o compartimento j fizer parte do problema a ser resolvido, ou seja, o compartimento teve novos itens atribuídos.

Para cada problema resolvido, a solução só é válida se atender aos quesitos de possuir todos os itens que já se encontravam no compartimento e carregar ao menos um item novo. As formulações matemáticas que representam a solução válida são descritas a seguir:

$$S^j \supset F^j \quad \forall j \quad (47)$$

$$S^j \cap A \neq \{\emptyset\} \quad \forall S^j \quad (48)$$

Após cada iteração, os valores de feromônios de atribuição dos itens aos compartimentos são atualizados levando em consideração a melhor solução da iteração para cada compartimento j , conforme o seguinte cálculo:

$$\tau_i^j \leftarrow (1 - e) \cdot \tau_i^j + \Delta\tau_i^j \cdot \left(p + \frac{a}{v}\right) \quad \forall i \in A \quad (49)$$

A cada iteração também é realizada a atualização dos conjuntos de itens. Os itens que foram carregados na melhor solução do problema do compartimento j , passa a fazer parte do conjunto F^j .

Finalmente, o problema será considerado resolvido quando todos os itens do conjunto A forem carregados, ou seja, o conjunto A estiver vazio.

4 TESTES COMPUTACIONAIS

Para verificação do desempenho da resolução do Problema do Empacotamento Tridimensional com rotação através do Algoritmo de Otimização da Colônia de Formigas e do Método de Geração de Colunas foram utilizadas as 320 instâncias criadas por Martello et. al. (2000). Estas instâncias podem ser geradas a partir de código em Linguagem C disponibilizado na página na Web do Pisinger (<http://hjemmesider.diku.dk/~pisinger/codes.html>).

Para geração destas instâncias, partiu-se de 5 tipos de itens que são gerados randomicamente com as dimensões entre o intervalo especificado na Tabela 3:

Tabela 3 – Intervalo de dimensões do item

Tipo	w_i	h_i	l_i
1	$[1, \frac{1}{2}W]$	$[\frac{2}{3}H, H]$	$[\frac{2}{3}L, L]$
2	$[\frac{2}{3}W, W]$	$[1, \frac{1}{2}H]$	$[\frac{2}{3}L, L]$
3	$[\frac{2}{3}W, W]$	$[\frac{2}{3}H, H]$	$[1, \frac{1}{2}L]$
4	$[\frac{1}{2}W, W]$	$[\frac{1}{2}H, H]$	$[\frac{1}{2}L, L]$
5	$[1, \frac{1}{2}W]$	$[1, \frac{1}{2}H]$	$[1, \frac{1}{2}L]$

Fonte: Martello et al., 2000.

Estes tipos de itens foram gerados para serem usados também em problemas sem rotação de itens. De fato, quando se rotaciona os itens e levando em consideração que $W = H = L$, o item tipo 1 é igual aos itens tipo 2 e tipo 3. Então temos apenas 3 tipos de itens para o problema com rotação.

Tabela 4 – Percentual de tipo de itens que compõe cada classe de problemas

Classe	L, W, H	Tipo de Item				
		1	2	3	4	5
1	100	60%	10%	10%	10%	10%
2	100	10%	60%	10%	10%	10%
3	100	10%	10%	60%	10%	10%
4	100	10%	10%	10%	60%	10%
5	100	10%	10%	10%	10%	60%

Fonte: Mahvash et. al., 2017.

Destes tipos de itens são criadas 8 classes de problema. O percentual de cada tipo de itens presentes em cada classe de problema encontra-se na Tabela 4:

A classe 6 tem compartimentos com dimensões L, W e H igual a 10 e itens distribuídos com dimensões geradas randomicamente entre 1 e 10. A classe 7 tem compartimentos com dimensões L, W e H igual a 40 e itens distribuídos com dimensões geradas randomicamente entre 1 e 35. A classe 8 tem compartimentos com dimensões L, W e H igual a 100 e itens distribuídos com dimensões geradas randomicamente entre 1 e 100.

As características das classes podem ser resumidas conforme a Tabela 5. Esta tabela apresenta como é a distribuição dos itens em cada classe de problema de acordo com os limites percentuais de volume dos itens em relação ao tamanho do compartimento.

Tabela 5 – Percentual de itens (por volume) que compõe cada classe de problemas

Classe	Limites dos Volume dos Itens em Percentuais do Tamanho da Bin				
	$\cong [1\%, 13\%]$	$\cong [1\%, 50\%]$	$\cong [3\%, 88\%]$	$\cong [1\%, 100\%]$	$\cong [10\%, 100\%]$
1, 2 e 3	10%	80%	-	-	10%
4	10%	30%	-	-	60%
5	60%	30%	-	-	10%
6	-	-	-	-	100%
7	-	-	100%	-	-
8	-	-	-	100%	-

Cada uma das 8 classes possui problemas de 50, 100, 150 e 200 itens e cada quantidade de itens possui 10 instâncias, totalizando as 320 instâncias.

Para os testes computacionais foi utilizado um computador com um processador de 2,1GHz. A codificação foi realizada em Linguagem C e foi utilizado o Software Gurobi para a resolução dos Problemas de Programação Linear e Programação Inteira.

4.1 HEURÍSTICA DE PONTOS EXTREMOS

Para determinar qual parâmetro seria utilizado como heurística, foi realizado testes com a Heurística de Pontos Extremos. Crainic et.al. (2008) realizaram testes com a Heurística em problemas sem rotação, utilizando as classes de problemas 1, 4, 5, 6, 7 e 8. Para as resoluções sem partição, chegou-se à conclusão de que a melhor estratégia era ordenar os itens prioritariamente pela maior área.

Tabela 6 – Comparação entre Desempenhos da Heurística de Pontos Extremos

Classe	Qtde	Heurística Pontos Extremos			
		Ordenado por Área	Ordenado por Maior Lado	Ordenado por Volume	Melhor Resultado (sem rotação)
1	50	13,7	13,7	12,7	13,8
	100	27,8	27,4	25,2	27,4
	150	38	38,1	35,2	37,7
	200	53	52,4	46,9	52,3
2	50	14,4	13,8	13,2	-
	100	27,3	26,2	24,3	-
	150	37,8	36,4	35	-
	200	51,5	48,5	46,7	-
3	50	14,1	14,1	12,8	-
	100	26,9	27,1	24,6	-
	150	38,8	38,9	35,9	-
	200	51,5	51,3	46,4	-
4	50	29,8	29,2	29	29,5
	100	59,2	59,1	58,5	59
	150	87,6	87,1	86,7	86,9
	200	119,4	119	118,5	118,9
5	50	8,7	8,5	8,3	8,4
	100	16,4	16	14,9	15,4
	150	22,5	21,9	20,2	21
	200	31	29,4	27,1	28,2
6	50	10,2	10,5	10	10,1
	100	19,9	20,7	19,7	19,8
	150	31,2	31,2	29,8	30,2
	200	40	40,8	38,4	38,6
7	50	7,5	7,9	7	7,6
	100	12,8	13,3	12,4	13,3
	150	16,9	17,4	16,2	16,9
	200	25,5	25,4	24	24,9
8	50	9,3	10,1	9	9,5
	100	19,3	20,1	18,5	19,4
	150	25,5	26,3	24,3	25,4
	200	30,7	32	30,1	31,4

Em testes realizados com a possibilidade de rotação, ficou demonstrada a vantagem em se classificar os itens por volume (Tabela 6). A última coluna apresenta os resultados obtidos por Crainic et.al. (2008) para o problema sem possibilidade de rotação.

Os padrões gerados pela Heurística de Pontos Extremos e utilizados como soluções finais nas meta-heurísticas deste trabalho foram validados ao final, certificando-se que os mesmos obedeciam às restrições de não ultrapassar os limites do compartimento (Fórmula de 14 a 16) e de não sobreposição dos itens (Fórmula de 17 a 22), presentes na Subseção 2.3.

4.2 MÉTODO DE GERAÇÃO DE COLUNAS

O Estado da Arte na resolução das instâncias de Martelo et. al. (2000) com rotação é o Método de Geração de Colunas aplicado por Mahvash et. al. (2017). Os autores codificaram em na linguagem C++ utilizando um computador de 2,2 GHz. O *solver* utilizado no Estado da Arte foi o CPLEX.

No presente trabalho foi realizada a implementação do método com as seguintes adaptações:

- I. Resolução do Problema Mestre com a restrição alterada conforme a fórmula (37'), para não haver problema *infeasible*, e tratamento da solução retirando os itens repetidos.
- II. Restrição de tempo conforme a Tabela 7, pois os autores informaram o tempo limite de 1000 minutos, mas não especificaram como este limite foi particionado entre o Problema Mestre Restrito e o Problema Mestre.

Tabela 7 - Limite tempo de execução para o Método de Geração de Colunas

Quantidade	Problema Mestre Restrito	Problema Mestre
50	30 segundos	970 segundos
100	120 segundos	880 segundos
150	150 segundos	850 segundos
200	180 segundos	820 segundos

4.3 ALGORITMO DE OTIMIZAÇÃO DA COLÔNIA DE FORMIGAS

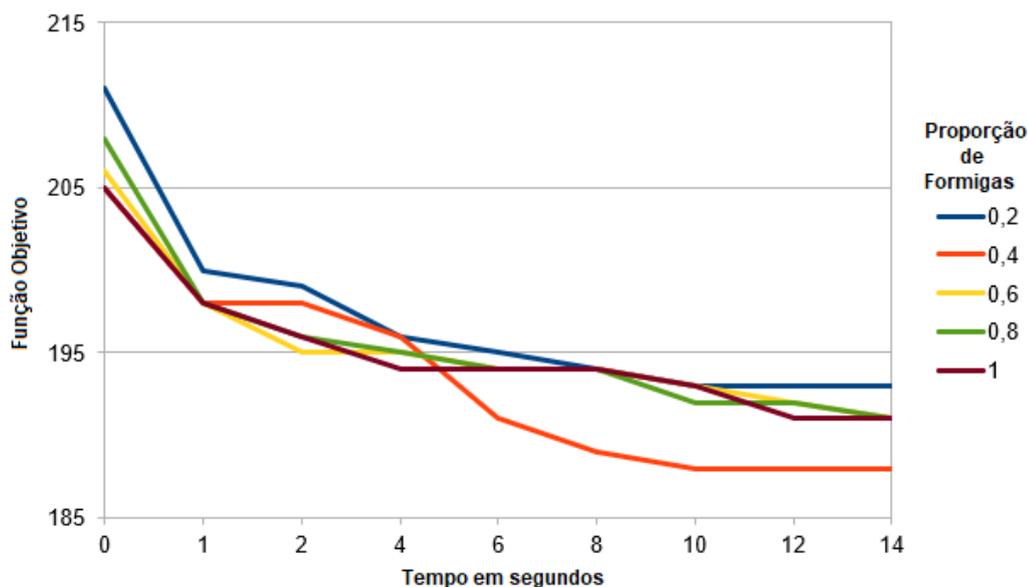
Para a definição dos parâmetros do Algoritmo de Otimização da Colônia de Formigas a serem utilizados nos problemas de 50 e 150 itens, foram realizados testes em uma amostra de 16 instâncias de 50 itens (2 instâncias de cada classe de

problema). Já a definição dos parâmetros para os problemas de 150 e 200 itens foi obtida através de testes em 8 instâncias de 150 itens (1 instância de cada classe de problema). Para as instâncias com 50 itens a execução foi limitada a 15 segundos, enquanto para as instâncias com 150 itens, o limite foi de 250 segundos.

Posteriormente, por razão de fraco desempenho do método na resolução dos problemas das classes de 1 a 4, foi necessário customizar os parâmetros por classe, e não por quantidade, para resolução dos problemas das referidas classes.

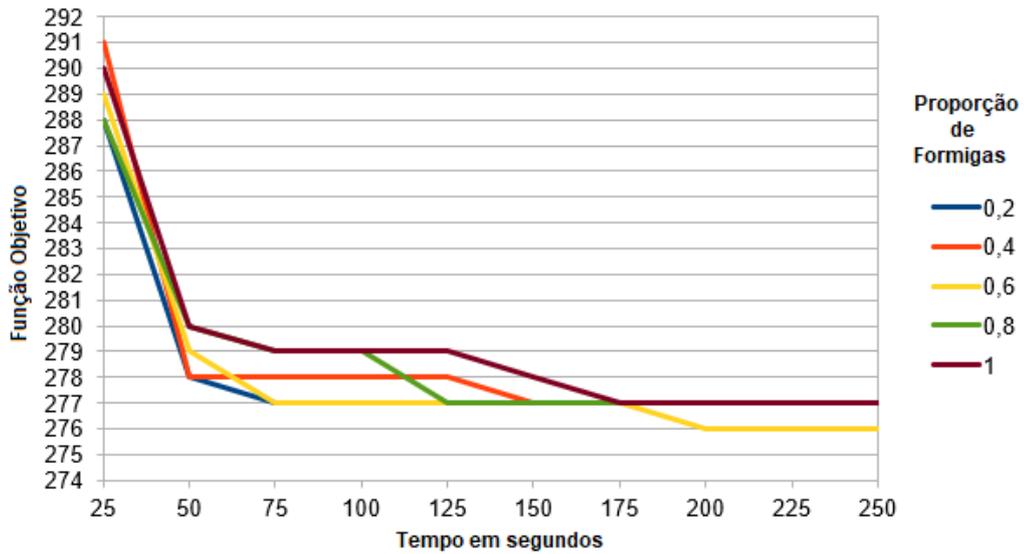
Nos testes por divididos por quantidade de itens, os gráficos 1 e 2 mostram a evolução da convergência do método ao longo do tempo para as amostras de instâncias com 50 e 150 itens respectivamente. Estes gráficos comparam alguns valores de Proporção de Formigas (Q) que compõe o cálculo da Fórmula 30.

Gráfico 1 – Proporção de Formigas (Q) em instâncias com 50 itens



Fonte: elaborado pelo autor

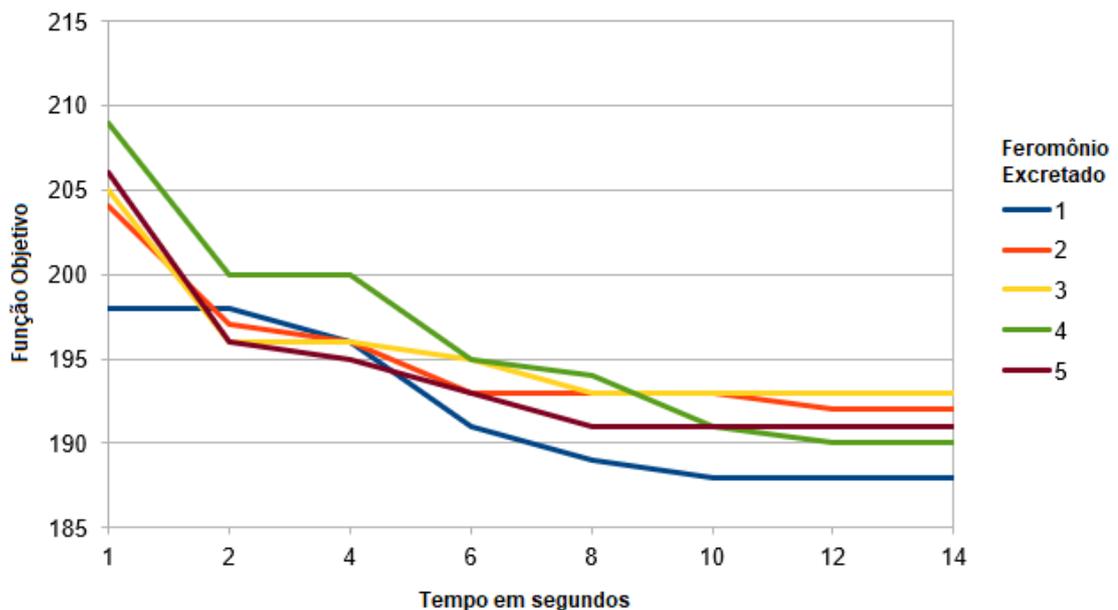
Gráfico 2 - Proporção de Formigas (Q) em instâncias com 150 itens



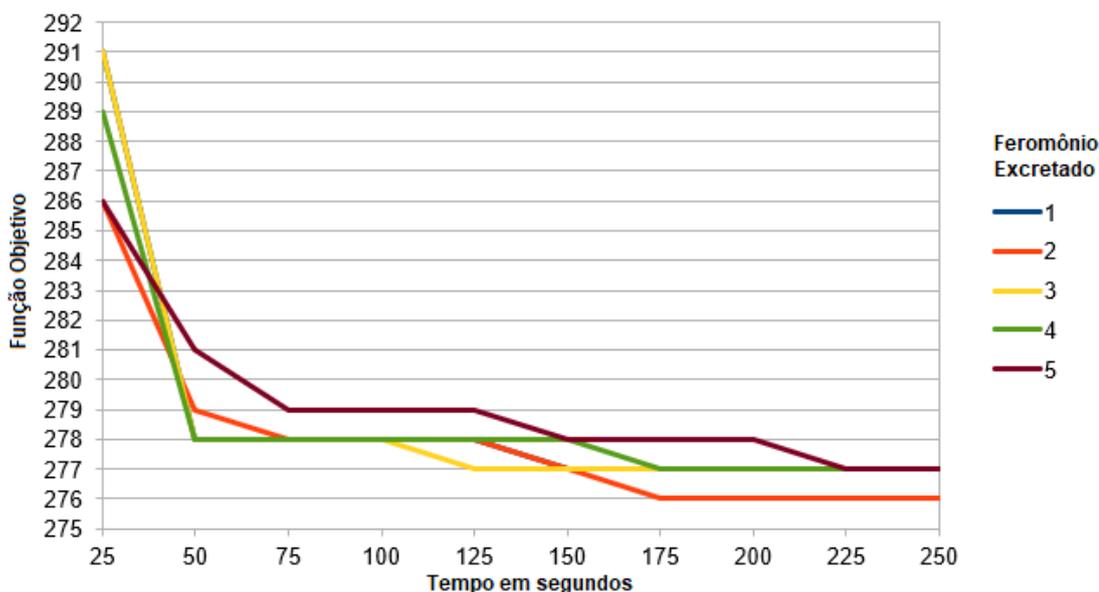
Fonte: elaborado pelo autor

Os gráficos 3 e 4 apresentam a evolução da convergência ao longo do tempo para as amostras de instâncias com 50 e 150 itens, respectivamente, e comparam alguns valores de Feromônio Excretado (p) que compõe o cálculo da Fórmula 35'.

Gráfico 3 – Feromônio excretado (p) em instâncias com 50 itens

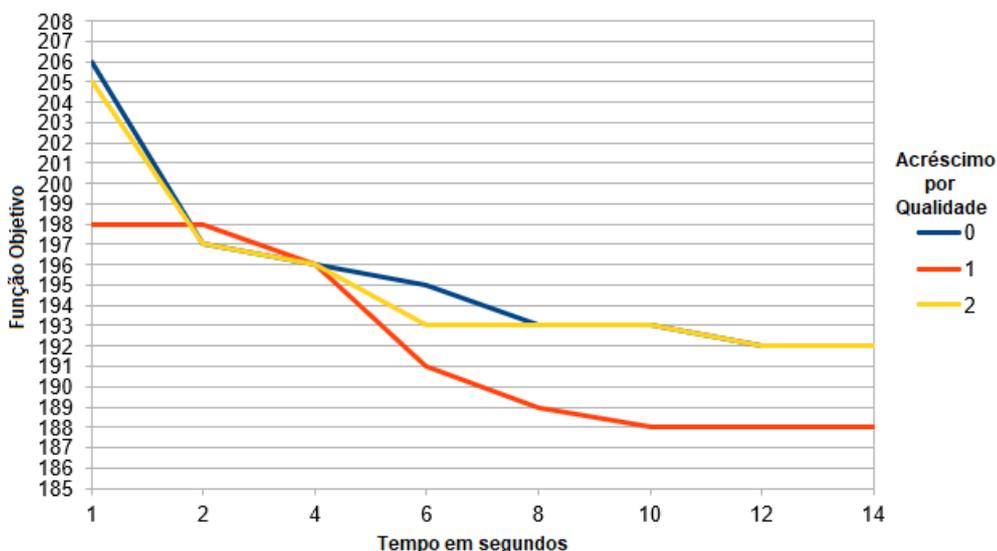


Fonte: elaborado pelo autor

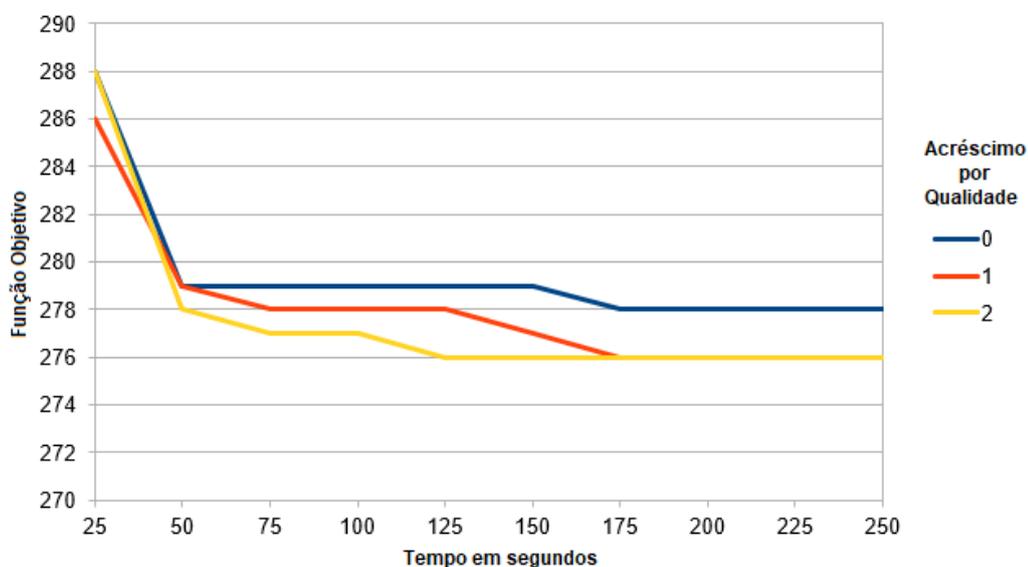
Gráfico 4 – Feromônio excretado (p) em instâncias com 150 itens

Fonte: elaborado pelo autor

Os gráficos 3 e 4 apresentam a evolução da convergência ao longo do tempo para as amostras de instâncias com 50 e 150 itens, respectivamente, e comparam alguns valores de Acréscimo por Qualidade (a) que compõe o cálculo da Fórmula 35'. Por estes gráficos é possível constatar que a utilização da técnica foi efetiva, pois o algoritmo performou melhor que quando não utilizada a técnica ($a = 0$).

Gráfico 5 – Acréscimo por Qualidade (a) em instâncias com 50 itens

Fonte: elaborado pelo autor

Gráfico 6 – Acréscimo por Qualidade (α) em instâncias com 150 itens

Fonte: elaborado pelo autor

Através destes gráficos é possível constatar a relação entre a velocidade e a qualidade da convergência, conforme abordado na Subseção 3.2.2.2. Parâmetros que inicialmente imprimem uma rápida convergência podem, ao longo da execução, ter resultados piores que parâmetros que apresentem uma curva mais suave de convergência.

Os parâmetros utilizados neste trabalho para as classes de problemas de 1 a 4 encontram-se na Tabela 8.

Tabela 8 - Valores de Parâmetros ACO (Classe 1 a 4)

Parâmetros	Classes 1 a 3	Classe 4
Q	1	1,5
Q_{η}	1	1,5
p	1	0,5
α	1	1
β	1	0,5
e	0,05	0,05
a	1	1
d	0,01	0,005

Os parâmetros utilizados neste trabalho para as classes de problemas de 5 a 8 encontram-se na Tabela 9.

Tabela 9 – Valores de Parâmetros ACO (Classe 5 a 8)

Parâmetros	50 e 100 itens	150 e 200 itens
Q	0,4	0,6
Q_η	1,5	1,5
p	1	1
α	1	1
β	1	2
e	0,05	0,05
a	1	2
d	0,01	0,005

4.4 COMPARAÇÃO ENTRE AS IMPLEMENTAÇÕES

A comparação entre as implementações, tomando como base o Método de Geração de Colunas encontra-se na Tabela 10.

Tabela 10 - Comparação de desempenho entre os Métodos Implementados

Classe	Qtde	CG		ACO		ACO + CG		CG + ACO	
		Objetivo	Tempo	Objetivo	Tempo	Objetivo	Tempo	Objetivo	Tempo
1	50	11,7	12	12	5	12,3	9	11,7	3
	100	23,2	48	24,4	24	23,9	29	23,4	13
	150	32,5	531	34,1	20	33,6	142	33	52
	200	44,9	771	46,1	244	45,9	213	45,5	70
2	50	12,2	10	12,5	5	12,6	9	12,2	2
	100	23	54	23,8	6	23,5	27	22,9	17
	150	32,4	357	34,3	65	33,6	145	33,1	52
	200	44,3	894	45,2	248	45,2	188	44,4	85
3	50	11,9	21	12,1	5	12,5	7	11,9	4
	100	22,9	38	23,8	8	23,5	27	23	12
	150	33,1	485	34,9	57	34,5	141	33,6	47
	200	44,3	876	45,5	99	45	213	44,5	70
4	50	28,9	1	28,9	1	28,9	1	28,9	1
	100	58,4	1	58,4	1	58,5	1	58,4	1
	150	86,4	1	86,4	1	86,4	1	86,4	1
	200	118,3	1	118,3	1	118,3	1	118,3	1
5	50	7,7	9	7,5	7	8	7	7,6	3
	100	14,2	38	14,4	27	14,8	15	14,2	13
	150	19,6	456	19,8	63	20,7	94	19,7	50
	200	26,9	639	27	130	27,8	163	26,9	73
6	50	9,6	12	9	7	9,7	7	9,6	3
	100	18,7	124	18,3	26	19,1	21	19,1	18
	150	28,8	510	28,3	58	29,1	133	29	42
	200	38	745	36,9	105	38,5	123	37,8	85
7	50	7	7	6,6	7	7,3	5	7	2
	100	12,4	166	11,8	26	12,8	9	12,2	15
	150	16,3	822	15	71	17,5	2	16,1	63
	200	24	902	23,3	123	25,5	31	23,9	95
8	50	8,7	12	8,4	8	9,2	1	8,6	3
	100	18	42	18,2	27	18,5	25	17,9	10
	150	23,7	932	23,5	67	24,8	74	23,8	57
	200	30,1	789	29,2	134	31,6	48	30,1	86
Total		9321	103060	9571	25900	9531	19120	9347	10490

A Tabela 11 apresenta, por classe de problemas, o *gap* médio que cada método apresentou em termos de função objetivo em relação ao método de comparação (GC). O método de Geração de Colunas combinado com o método de Colônia de Formigas performou melhor na resolução dos problemas de classe 1, 2 e

3, enquanto a Colônia de Formigas apresentou melhor desempenho na resolução dos problemas de classe 6, 7 e 8.

Tabela 11 – Gap Médio do Valor da Função Objetivo em Relação ao GC por Classe de Problema

Classe	ACO	ACO + CG	CG + ACO
1	4%	3%	1%
2	3%	3%	0%
3	3%	3%	1%
4	0%	0%	0%
5	0%	4%	0%
6	-3%	1%	1%
7	-5%	5%	-1%
8	-2%	5%	0%

A Tabela 12 registra, por classe de problema, o *gap* médio de tempo em relação ao tempo de execução do Método de Geração de Colunas. A combinação Método de Geração de Colunas com Algoritmo de Otimização da Colônia de Formigas obteve as melhores médias de tempo. O destaque, no entanto, ficou para o Método de Colônia de Formigas, que apresentou bons resultados de função objetivo para as classes 6, 7 e 8 em menor tempo médio que o método de comparação (Geração de Colunas).

Tabela 12 – Gap Médio de Tempo de Execução em Relação ao CG por classe de problema

Classe	ACO	ACO + CG	CG + ACO
1	-68%	-54%	-84%
2	-73%	-51%	-82%
3	-83%	-61%	-84%
4	0%	0%	0%
5	-54%	-62%	-80%
6	-74%	-71%	-87%
7	-66%	-82%	-88%
8	-71%	-82%	-85%

A Tabela 13 apresenta, por quantidade de itens, o *gap* médio em termos de função objetivo em relação ao GC. O Algoritmo de Otimização da Colônia de Formigas apresentou melhores desempenhos na resolução de instâncias com 50 e 100 itens e desempenho similar à combinação dos métodos CG + ACO para as instâncias de 150 e 200 itens.

Tabela 13 – Gap Médio do Valor da Função Objetivo em Relação ao CG por quantidade de itens

Qtde Itens	ACO	ACO + CG	CG + ACO
50	-1%	4%	0%
100	1%	3%	0%
150	1%	4%	1%
200	0%	3%	0%

A Tabela 14 registra, por quantidade de itens da instância, o *gap* médio de tempo em relação ao tempo de execução do Método de Geração de Colunas. A combinação Método de Geração de Colunas com Algoritmo de Otimização da Colônia de Formigas se destacou na resolução de instâncias de 100, 150 e 200 itens, ao performar melhor em menor percentual de tempo médio. Na resolução de problemas com 50 itens, embora não tenha tido o mesmo desempenho do CG + ACO, o Método de Colônia de Formigas se destacou ao apresentar bons resultados de função objetivo em menor tempo médio que o método de comparação (Geração de Colunas).

Tabela 14 – Gap Médio de Tempo de Execução em Relação ao CG por quantidade de itens

Qtde Itens	ACO	ACO + CG	CG + ACO
50	-35%	-41%	-70%
100	-56%	-50%	-67%
150	-78%	-69%	-79%
200	-71%	-72%	-79%

A combinação CG + ACO apresentou bom desempenho na resolução de problemas das classes de 1 a 4, enquanto o ACO demonstrou bom desempenho para nos problemas de classe 4 a 8.

4.5 COMPARAÇÃO COM ESTADO DA ARTE

A Tabela 15 apresenta a comparação entre a implementação dos quatro métodos neste trabalho com o Estado da Arte apresentado em Mahvash et. al. (2017).

Tabela 15 – Comparação de Desempenho com o Estado da Arte

Classe	Qtde	CG		ACO		ACO + CG		CG + ACO		CG Mashvash et. al., 2017	
		Obj	Tempo	Obj	Tempo	Obj	Tempo	Obj	Tempo	Obj	Tempo
1	50	11,7	12	12	5	12,3	9	11,7	3	11,7	10
	100	23,2	48	24,4	24	23,9	29	23,4	13	22,9	60
	150	32,5	531	34,1	20	33,6	142	33	52	31,3	200
	200	44,9	771	46,1	244	45,9	213	45,5	70	42,8	1000
2	50	12,2	10	12,5	5	12,6	9	12,2	2	11,8	10
	100	23	54	23,8	6	23,5	27	22,9	17	22,4	70
	150	32,4	357	34,3	65	33,6	145	33,1	52	31,4	100
	200	44,3	894	45,2	248	45,2	188	44,4	85	41,9	1000
3	50	11,9	21	12,1	5	12,5	7	11,9	4	11,8	9
	100	22,9	38	23,8	8	23,5	27	23	12	22,5	60
	150	33,1	485	34,9	57	34,5	141	33,6	47	32,0	300
	200	44,3	876	45,5	99	45	213	44,5	70	42,1	1000
4	50	28,9	1	28,9	1	28,9	1	28,9	1	28,9	1
	100	58,4	1	58,4	1	58,5	1	58,4	1	58,4 ³	3
	150	86,4	1	86,4	1	86,4	1	86,4	1	86,4	4
	200	118,3	1	118,3	1	118,3	1	118,3	1	118,3	17
5	50	7,7	9	7,5	7	8	7	7,6	3	7,5	5
	100	14,2	38	14,4	27	14,8	15	14,2	13	13,7	100
	150	19,6	456	19,8	63	20,7	94	19,7	50	18,6	400
	200	26,9	639	27	130	27,8	163	26,9	73	25,3	1000
6	50	9,6	12	9	7	9,7	7	9,6	3	8,9	5
	100	18,7	124	18,3	26	19,1	21	19,1	18	17,9	100
	150	28,8	510	28,3	58	29,1	133	29	42	27,5	200
	200	38	745	36,9	105	38,5	123	37,8	85	35,5	1000
7	50	7	7	6,6	7	7,3	5	7	2	6,6	16
	100	12,4	166	11,8	26	12,8	9	12,2	15	11,2	200
	150	16,3	822	15	71	17,5	2	16,1	63	14,3	500
	200	24	902	23,3	123	25,5	31	23,9	95	21,9	1000
8	50	8,7	12	8,4	8	9,2	1	8,6	3	8,3	10
	100	18	42	18,2	27	18,5	25	17,9	10	17,7	22
	150	23,7	932	23,5	67	24,8	74	23,8	57	22,1	500
	200	30,1	789	29,2	134	31,6	48	30,1	86	27,7	1000
Total		9321	10306	9571	2590	9531	1912	9347	1049	9013	9902

Em valores absolutos, o Método de Geração de Colunas utilizados por Mahvash et al. (2017) apresentou melhores resultados em termos de função objetivo. Nem mesmo a replicação do método de Geração de Colunas conseguiu chegar aos valores apresentados por Mahvash et al. (2017). Algumas hipóteses para não se ter alcançado os valores são:

- i. A implementação da Heurística de Pontos Extremos não ter sido tão eficiente quanto a implementação da literatura; e
- ii. Os parâmetros de tempo (divisão do tempo entre o Problema Mestre Restrito e o Problema mestre) diferir da literatura, já que foi informado apenas o tempo limite total.

Desconsiderando o Método de Geração de Colunas, que é uma replicação do método apresentado na literatura, a combinação CG + ACO foi a que mais se aproximou dos valores encontrados na literatura para as classes 1, 2, 3, 4 e 5. Para as classes 6, 7 e 8, o método que mais se aproximou dos valores registrados na literatura foi o Algoritmo de Otimização da Colônia de Formigas.

A Tabela 16 demonstra as diferenças percentuais entre os métodos implementados neste trabalho em comparação com o Estado da Arte.

Tabela 16 –Gap Médio do Valor da Função Objetivo para o Estado da Arte por classe de problema

Classe	CG	ACO	ACO + CG	CG + ACO
1	3%	6%	6%	3%
2	4%	7%	7%	4%
3	3%	6%	6%	3%
4	0%	0%	0%	0%
5	5%	5%	9%	4%
6	6%	3%	7%	7%
7	10%	4%	16%	9%
8	6%	4%	10%	5%

Em relação à média do percentual de tempo em relação ao Método de Geração de Colunas utilizados por Mahvash et al. (2017), a combinação dos Métodos Geração de Colunas e Colônia de Formigas apresentou performances de tempo bem abaixo das registradas na literatura (Tabela 16), com uma boa aproximação dos valores de função objetivo (Tabela 15) para as classes de problema de 1 a 5. Já o Algoritmo de Otimização da Colônia de Formigas apresentou tempos de execução significativamente menores que os registrados na literatura (Tabela 17) com boa aproximação dos valores de função objetivo (Tabela 16) para as classes de problema 6, 7 e 8.

Tabela 17 – Gap Médio de Tempo de Execução para o Estado da Arte por classe de problema

Classe	CG	ACO	ACO + CG	CG + ACO
1	36%	-69%	-42%	-79%
2	56%	-63%	-27%	-74%
3	36%	-76%	-52%	-78%
4	-59%	-59%	-59%	-59%
5	-1%	-51%	-51%	-77%
6	73%	-49%	-40%	-73%
7	-5%	-79%	-90%	-89%
8	44%	-43%	-64%	-76%

Desconsiderando o Método de Geração de Colunas, que é uma replicação do método apresentado na literatura, a combinação CG + ACO foi a que mais se aproximou dos valores encontrados na literatura nos grupamentos por quantidade de itens (Tabela 18).

Tabela 18 – Gap Médio do Valor da Função Objetivo para o Estado da Arte por quantidade de itens

Qtde Itens	CG	ACO	ACO + CG	CG + ACO
50	3%	3%	7%	3%
100	3%	4%	6%	3%
150	5%	6%	9%	6%
200	6%	6%	9%	6%

Em relação ao gap médio do tempo de execução, a combinação dos Métodos Geração de Colunas e Colônia de Formigas apresentou performances de tempo significativamente abaixo das registradas na literatura (Tabela 18), obtendo boas aproximações de resultados (Tabela 17).

Tabela 19 – Gap Médio de Tempo de Execução em Relação ao Estado da Arte por quantidade de itens

Qtde Itens	CG	ACO	ACO + CG	CG + ACO
50	42%	-18%	-15%	-55%
100	-14%	-65%	-60%	-77%
150	91%	-76%	-51%	-78%
200	-29%	-86%	-87%	-92%

De forma geral, a replicação do método GC foi o que mais aproximou do Estado da Arte, tanto em valores de função objetivo (3,4% de *gap*), quanto em tempo de execução (4,1% de *gap*), embora não tenha conseguido alcançar os valores de função objetivo registrados na literatura. O método que combina Geração

de Colunas e Colônia de Formigas apresentou o melhor desempenho entre os métodos desenvolvidos neste trabalho, obtendo 3,7% de *gap* geral em relação ao Estado da Arte com um *gap* de -89,4% de tempo de execução. O método baseado no Algoritmo de Otimização da Colônia de Formigas apresentou o segundo melhor desempenho, com um *gap* geral de 4,1% no valor de função objetivo e -83,1% em de tempo de execução. Por último ficou o método baseado em ACO + GC, que apresentou um *gap* geral de 5,7% no valor da função objetivo e 80,7% no tempo de execução.

Ficou demonstrado que o Método de Geração de Colunas é eficaz na obtenção de bons resultados, o que se deve principalmente a capacidade de não descartar e combinar as várias soluções geradas durante a resolução do problema. Por outro lado, apresenta pior desempenho no tempo de execução em relação aos métodos desenvolvidos neste trabalho, podendo ser atribuído a isto a demora na resolução do Problema Master, por se tratar de um Problema de Otimização Inteira.

5 CONCLUSÃO

Neste trabalho, foram implementados quatro métodos de resolução do Problema do Empacotamento Tridimensional com rotação com a intenção de se obter soluções em menor tempo, principalmente para as maiores instâncias em relação à literatura. O Método de Geração de Colunas implementado foi a replicação do Estado da Arte, base de comparação para os outros três métodos desenvolvidos baseados em: Algoritmo de Otimização da Colônia de Formigas; combinação do Algoritmo de Otimização da Colônia de Formigas e Método de Geração de Colunas; e combinação do Método de Geração de Colunas e Algoritmo de Otimização da Colônia de Formigas.

De modo geral, não foram alcançados os valores de função objetivo da implementação de Mahvash et. al. (2017) nem mesmo na replicação do método. Isto pode ser em razão de ineficiência da implementação da Heurística de Pontos Extremos ou do racionamento do limite de tempo total entre o Problema Mestre Restrito e o Problema Mestre. Por outro lado, foram obtidas boas soluções, com pequenos *gaps* às melhores soluções conhecidas, atingindo o principal objetivo do trabalho.

Ficou demonstrado, tanto na literatura, quanto na replicação do método, que o Método de Geração de Colunas consegue obter melhores valores de função objetivo, mas tem um desempenho de tempo não tão eficiente quanto aos métodos desenvolvidos neste trabalho.

De modo geral, excluindo a replicação do Estado da Arte, a combinação dos Método de Geração de Colunas e o Algoritmo de Otimização da Colônia de Formigas conseguiu aproximar mais dos melhores valores de função objetivo obtidos na literatura. O segundo método que mais conseguiu se aproximar foi o baseado na Colônia de Formigas. Ambos os métodos conseguiram bons desempenho de função objetivo em um tempo significativamente mais rápido que os obtidos no método de referência.

Para as classes de problema de 1 a 5, a combinação do Método de Geração de Colunas com o Algoritmo de Otimização da Colônia de Formigas obteve um Gap Médio de 2,8% em um tempo médio de execução com um *gap* médio de -73,4%. O Algoritmo de Otimização da Colônia de Formigas performou bem para as classes de

problema 6, 7 e 8, conseguindo uma média de 3,7% de *Gap* médio, em um gap médio de tempo de -57%. Por se tratar de um problema operacional, onde pode ser exigido um tempo rápido de resposta, a ganho no tempo de execução é importante.

Para trabalhos futuros recomenda-se uma revisão da Heurística de Pontos Extremos, a realização de testes com diferentes limites de tempo para o Método de Geração de Colunas quanto a divisão do tempo total de execução entre o Problema Mestre Restrito e o Problema Mestre nos métodos baseados em Geração de Colunas. Ainda nos métodos baseados em Geração de Colunas, recomenda-se a adoção de técnicas que possam diminuir o tempo de execução do Problema Mestre, podendo ser utilizado, por exemplo, técnicas de seleção de colunas. Além disso, presente-se expandir os métodos desenvolvidos neste trabalho para aplicação em variantes do clássico PET, que incorporam aspectos mais realistas do problema, como a consideração de restrições de transporte, de equilíbrio estrutural e de integração das rotas de entrega dos contêineres.

BIBLIOGRAFIA

ALI, Sara; RAMOS, António Galvão; CARRAVILLA, Maria Antónia; OLIVEIRA, José Fernando. **On-line three-dimensional packing problems: A review of off-line and on-line solution approaches**. Computers and Industrial Engineering, 168, p. 108-122, 2022.

ALONSO, M.T.; ALVAREZ-VALDES, R.; PARREÑO, F. A. **GRASP algorithm for multi container loading problems with practical constraints**. 4OR-Q A Quarterly Journal of Operations Research, 18, p. 49–72, 2020.

ARENALES, Marcos N.; MORABITO, Reinaldo; YANASSE, Horácio Hideki. **Problemas de Corte e Empacotamento**. Livro-texto do Mini curso, XXVI Simpósio Brasileiro de Pesquisa Operacional, São João Del Rei, MG - Brasil, 2004.

BARNHART, Cynthia; JOHNSON, Ellis L.; NEMHAUSER, George L.; SAVELSBERGH, Martin W. P.; VANCE, Pamela H. **Branch-and-Price: Column Generation for Solving Huge Integer Programs**. Operations Research 46.3, p. 316-329, 1998.

BOSCHETTI, Marco A. **New lower bounds for the three-dimensional finite bin packing problem**. Discrete Applied Mathematics, 140, p. 241-258, 2004.

CHEN, C. S.; LEE, M. S.; SHEN, Q. S. **An analytical model for the container loading problem**. European Journal of Operational Research, v. 80, p. 68-78, 1995.

CONFEDERAÇÃO NACIONAL DA INDÚSTRIA. **Revista Indústria Brasileira**, ano 6, nº 62, 2021. Disponível em https://jornalismo.portaldaindustria.com.br/cni/revista_industria/revista-industria-brasileira-12-2021/. Acesso em 12 maio 2022.

CRAINIC, Teodor Gabriel; PERBOLI, Guido; TADEI, Roberto. **Extreme Point-Based Heuristics for Three-Dimensional Bin Packing**. Inform Journal on Computing, 20 - 3, p. 368-384, 2008.

CRAINIC, Teodor Gabriel; PERBOLI, Guido; TADEI, Roberto. **TS² PACK: A two-level tabu search for the three-dimensional bin packing problem**. European Journal of Operational Research, 195, p. 744-760, 2009.

DORIGO, Marco; BLUM, Christian. **Ant colony optimization theory: A survey**. Theoretical Computer Science, 344, p. 243-278, 2005.

DORIGO, Marco; GAMBARDELLA, Luca Maria. **Ant colonies for travelling salesman problem**. BioSystems, 43, p. 73-81, 1997.

DUAN, Haibin; MA, Guanjun; LIU, Seqi. **Experimental Study of the Adjustable Parameters in Basic Ant Colony Optimization Algorithm**. IEEE Congress on Evolutionary Computation, p. 149-156, 2007.

EGEBLAD, Jens; PISINGER, David. **Heuristic approaches for two - and three-dimensional knapsack packing problem**. Computers and Operations Research, 36, p. 1026-1049, 2009.

FANSLAU, Tobias; BORTFELDT, Andreas. **A Tree Search Algorithm for Solving the Container Loading Problem**. Informs Journal on Computing, 22, p. 222-235, 2010.

FAROE, Oluf; PISINGER, David. ZACHARIASEN, Martin. **Guided Local Search for the Three-Dimensional Bin-Packing Problema**. Informs Journal on Computing, 15 (3), p. 267-283, 2003.

FENG, Xuehao; MOON, Ilkyeong; SHIN, Jeongho. **Hybrid genetic algorithms for the three-dimensional multiple container packing problem**. Flexible services and manufacturing journal, 27 (2-3), p. 451-457, 2013.

GILMORE, P. C.; GOMORY, R. E.. **A linear programming approach to the cutting-stock problem**. Operations Research. 10.1287/opre.11.6.863, 1963.

GILMORE, P. C.; GOMORY, R. E. **Multistage Cutting Stock Problems of Two and More Dimensions**. Operations Research. 13.1, p. 94-120, 1965.

GONÇALVES, José Fernando; RESENDE, Mauricio G. C. **A biased random key genetic algorithm for 2D and 3D bin packing problems**. International Journal of Production Economics, 145 (2), p. 500-510, 2013.

GZARA, Fatma; ELHEDHLI, Samir; YILDIZ, Burak C. **The Pallet Loading Problem: Three-dimensional bin packing with practical constraints**. European Journal of Operational Research, 287, p. 1062-1074, 2020.

KANG, Kyungdaw; MOON, Ilkyeong; WANG, Hongfeng. **A hybrid genetic algorithm with a new packing strategy for the three-dimensional bin packing problem**. Applied Mathematics and Computation, 219, p. 1287-1299, 2012.

KURPEL, Deidson Vitório. **Um modelo matemático para o problema de carregamento de contêineres**. Curitiba: Universidade Federal do Paraná, 2015.

KURPEL, Deidson Vitório; SCARPIN, Cassius Tadeu; JUNIOR, José Eduardo Pécora; SCHENEKEMBERGER, Cleder Marcos; COELHO, Leandro C. **The exact solutions of several types of container loading problems**. European Journal of Operational Research, 284, p. 27-107, 2020.

LARA, Oscar D; LABRADOR, Miguel A. **A Multiobjective Ant Colony-Based Optimization Algorithm for Bin Packing Problem with Load Balancing**. IEEE Congress on Evolutionary Computation. p. 1-8, 2010.

LEVINE, John; DUCATELLE, Frederick. **Ant colony optimisation and local search for bin-packing and cutting stock problems**. Journal of the Operational Research Society, 55 (7), p. 705-716, 2004.

LIANG, Shyi-Ching; LEE, Chi-Yu; HUANG, Shih-Wei. **A Hybrid Metaheuristic for the Container Loading Problem**. Communications of the International Information Management Association, 7 (4), p. 73-84, 2007.

LÜBBECKE, Marco E.; DESROSIERS, Jacques. **Selected Topics in Column Generation**. Operations Research, 53 (6), p. 1007-1023, 2005.

MAHVASH, Batoul; AWASTHI, Anjali; CHAUHAN, Satyaveer. **A Column generation-based heuristic for the three-dimensional bin packing problem with rotation**. Journal of Operational Research Society, v. 69 (1), p. 78-90, 2017.

MARTELLO, Silvano; PISINGER, David; VIGO, Daniele. **The Three-Dimensional Bin Packing Problem**. Operations Research, 48(2), p. 256-267, 2000.

NASCIMENTO, Olivia Xavier do; QUEIROZ, Tiago Alves de; JUNQUEIRA, Leonardo. **Practical Constraint in container loading problem: Comprehensive formulations and exact algorithm**. Computers and Operations Research, 128, 2021.

OGUNRANTI, Gbemileke A.; OLULEYE, Ayodeji E. **Minimizing Waste (Off-cuts) Using Cutting Stock Model: The Case of One Dimensional Cutting Stock Problem in Wood Working Industry**. Journal of Industrial Engineering and Management, 9 (3), p. 834-859, 2016.

PARREÑO, F ; ALVAREZ-VALDES, R ; OLIVEIRA, J. F ; TAMARIT, J. M. **A hybrid GRASP/VND algorithm for two- and three-dimensional bin packing**. Computers & Applied Sciences Complete Annals of operations research, 179 (1), p. 203-220, 2008.

SCHEITHAUER, Guntram. **Algorithms for the Container Loading Problem**. Operations Research Proceedings, 1991, p. 445-452, 1991.

SHENG, Liu; WEI, Tan; ZHIYUAN, Xu; XIWEI, Liu. **A tree search algorithm for container loading problem**. Computers and Industrial Engineering, 75, p. 20-30, 2014.

SILVA, Everton Fernandes; TOFFOLO, Túlio Angelo Machado; WAUTERS, Tony. **Exact Methods for three-dimensional cutting and packing: A comparative study concerning single container problems**. Computers and Operations Research, 109, p. 12-27, 2019.