

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

Marcos Vinícius Almeida Guimarães

**UM MODELO EFICIENTE PARA O PROBLEMA DE CARREGAMENTO DE
CONTÊINERES**

Timóteo

2015

Marcos Vinícius Almeida Guimarães

**UM MODELO EFICIENTE PARA O PROBLEMA DE CARREGAMENTO DE
CONTÊINERES**

Monografia apresentada ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Engenheiro de Computação.

Orientador: André Rodrigues da Cruz

Timóteo

2015

Marcos Vinícius Almeida Guimarães

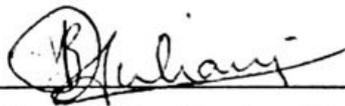
UM MODELO EFICIENTE PARA O PROBLEMA DE CARREGAMENTO DE CONTÊINERES

Monografia apresentada ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Engenheiro de Computação.

Orientador: André Rodrigues da Cruz



André Rodrigues da Cruz (Orientador)



João Batista Queiroz Zuliani



Júlio César de Jesus Onofre

Timóteo, 15 de dezembro de 2015

Agradeço aos meus pais pelo carinho, apoio e incentivo durante a graduação.

AGRADECIMENTOS

Aos meus pais, irmãos e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida. Mãe, seu cuidado e dedicação foram o que me deram, em alguns momentos, a esperança para seguir em frente. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada. Amo vocês.

Aos meus amigos Beatriz, Débora, Eder, Gabriel, Matheus A., Matheus C., Priscila, Ramon, Rodrigo M., Sebastião, Tamires, Tássio, Wallace e Warley, com os quais pude compartilhar momentos de felicidade, tristeza e desespero e que muito me ajudaram ao longo dessa trajetória. Cujas amizades pretendo levar por toda minha vida.

Especialmente à Rainara que sempre esteve do meu lado, sempre me apoiou em todas as decisões que eu tive que tomar. Sempre acreditou no meu potencial, me incentivando a ir cada vez mais além e conquistar os meus sonhos. Sua amizade é muito importante para mim e sem você, eu não estaria aqui. Eu posso dizer que a minha formação, inclusive pessoal, não teria sido a mesma sem a sua pessoa.

Ao meu orientador André, pela paciência na orientação e incentivo que tornaram possível a conclusão deste trabalho. Por estar sempre disposto a esclarecer as minhas dúvidas e me ajudar da melhor forma possível.

Aos professores João e Júlio, que aceitaram fazer parte da banca e por se disponibilizarem a avaliar o trabalho desenvolvido apontando a visão matemática compreendida dentro do problema tratado.

À professora Deismar, que muito me ajudou e orientou durante essa trajetória, esclarecendo as minhas dúvidas e apontando o caminho certo a ser seguido.

Ao CEFET-MG e a todos os professores do curso, que foram tão importantes na minha vida acadêmica e que muito contribuíram para a minha formação.

*“Do not pity the dead, Harry. Pity the living, and, above all,
those who live without love”.*

J. K. Rowling

RESUMO

A resolução do problema de carregamento de contêineres busca o melhor posicionamento de caixas dentro de quaisquer dispositivos de unitização de carga. Desde meados dos anos 60, várias alternativas foram criadas com o intuito de utilizar técnicas de Pesquisa Operacional para encontrar uma melhor solução. Muitas restrições podem ser levadas em consideração quando se trata do problema de carregamento de contêineres, dentre elas encontram-se, por exemplo, não-sobreposição das caixas, a estabilidade e prioridade do carregamento, a resistência das caixas ao serem empilhadas, o agrupamento e separação de itens, as considerações de múltiplos destinos e ordem de retirada, dentre outras. Na literatura técnica existente, tem-se um modelo muito utilizado de programação binária para gerar um conjunto de soluções viáveis para determinadas situações. Neste trabalho foi desenvolvido um novo modelo para resolver esse problema, de uma maneira mais eficiente, através da utilização de menos restrições e variáveis, acarretando em um tempo de execução e requisitos computacionais menores que o demonstrado na literatura. Este trabalho considerou o problema do carregamento de contêineres com restrições de não sobreposição.

Palavras-chave: Modelo matemático. Carregamento de contêineres. Programação Linear Inteira Mista. Pesquisa Operacional.

ABSTRACT

The solution to the container loading problem seeks the best way to position boxes within any load unitization devices. Since the mid-sixties several alternatives were created with the intention of using operations research techniques to find the best solution. Several constraints can be taken in consideration when it comes to the container loading problem, among them are, for example, non-overlapping boxes, loading stability and priority, resistance of the boxes to be stacked, grouping and separating items, considerations of multiple destinations and removal order, among others. In literature, a widely used model to generate a set of feasible solutions for certain situations shown. In this paper, a new model to solve this problem in a more efficient way is discussed, through the use of fewer constraints and variables, using less execution time and computational requirements than the one shown in literature. In this model, only restrictions that do not allow boxes to overlap each others are taken into consideration.

Keywords: Mathematical modeling. Container loading. Mixed Integer Linear Programming. Operations Research.

LISTA DE FIGURAS

FIGURA 1 –	Fases de um estudo de Pesquisa Operacional.....	13
FIGURA 2 –	Exemplo de posicionamento de uma caixa do tipo i dentro do contêiner.....	20
FIGURA 3 –	Sobreposição entre duas caixas	22
FIGURA 4 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 1 da primeira etapa de testes.....	31
FIGURA 5 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 2 da primeira etapa de testes.....	32
FIGURA 6 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 3 da primeira etapa de testes.....	33
FIGURA 7 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 4 da primeira etapa de testes.....	34
FIGURA 8 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 5 da primeira etapa de testes.....	35
FIGURA 9 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 6 da primeira etapa de testes.....	36
FIGURA 10 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 1 da segunda etapa de testes.....	39
FIGURA 11 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 2 da segunda etapa de testes.....	40
FIGURA 12 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 3 da segunda etapa de testes.....	41
FIGURA 13 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 4 da segunda etapa de testes.....	42
FIGURA 14 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 5 da segunda etapa de testes.....	43
FIGURA 15 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 6 da segunda etapa de testes.....	44
FIGURA 16 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 1 da terceira etapa de testes.....	46
FIGURA 17 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 2 da terceira etapa de testes.....	47

FIGURA 18 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 3 da terceira etapa de testes.....	48
FIGURA 19 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 4 da terceira etapa de testes.....	49
FIGURA 20 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 5 da terceira etapa de testes.....	50
FIGURA 21 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 6 da terceira etapa de testes.....	51
FIGURA 22 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 1 da quarta etapa de testes.....	53
FIGURA 23 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 2 da quarta etapa de testes.....	54
FIGURA 24 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 3 da quarta etapa de testes.....	55
FIGURA 25 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 4 da quarta etapa de testes.....	56
FIGURA 26 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 5 da quarta etapa de testes.....	57
FIGURA 27 –	Relação entre o tempo de execução e o valor final da função objetivo no Caso 6 da quarta etapa de testes.....	58

LISTA DE TABELAS

TABELA 1 –	Considerações apresentadas por Bischoff e Ratcliff (1995)	17
TABELA 2 –	Número total de variáveis e restrições que cada modelo utiliza no pior caso.....	28
TABELA 3 –	Tipos de caixas e suas respectivas dimensões	29
TABELA 4 –	Quantidade de tipos de caixa relacionada a cada caso da primeira etapa de testes.....	30
TABELA 5 –	Dados coletados no Caso 1 da primeira etapa de testes.....	31
TABELA 6 –	Dados coletados no Caso 2 da primeira etapa de testes.....	32
TABELA 7 –	Dados coletados no Caso 3 da primeira etapa de testes.....	33
TABELA 8 –	Dados coletados no Caso 4 da primeira etapa de testes.....	34
TABELA 9 –	Dados coletados no Caso 5 da primeira etapa de testes.....	35
TABELA 10 –	Dados coletados no Caso 6 da primeira etapa de testes.....	36
TABELA 11 –	Quantidade de tipos de caixa relacionada a cada caso da segunda etapa de testes.....	38
TABELA 12 –	Dados coletados no Caso 1 da segunda etapa de testes.....	38
TABELA 13 –	Dados coletados no Caso 2 da segunda etapa de testes.....	39
TABELA 14 –	Dados coletados no Caso 3 da segunda etapa de testes.....	40
TABELA 15 –	Dados coletados no Caso 4 da segunda etapa de testes.....	41
TABELA 16 –	Dados coletados no Caso 5 da segunda etapa de testes.....	42
TABELA 17 –	Dados coletados no Caso 6 da segunda etapa de testes.....	43
TABELA 18 –	Quantidade de tipos de caixa relacionada a cada caso da terceira etapa de testes.....	45
TABELA 19 –	Dados coletados no Caso 1 da terceira etapa de testes.....	45
TABELA 20 –	Dados coletados no Caso 2 da terceira etapa de testes.....	46
TABELA 21 –	Dados coletados no Caso 3 da terceira etapa de testes.....	47
TABELA 22 –	Dados coletados no Caso 4 da terceira etapa de testes.....	48
TABELA 23 –	Dados coletados no Caso 5 da terceira etapa de testes.....	49
TABELA 24 –	Dados coletados no Caso 6 da terceira etapa de testes.....	50
TABELA 25 –	Quantidade de tipos de caixa relacionada a cada caso da quarta etapa de testes.....	52
TABELA 26 –	Dados coletados no Caso 1 da quarta etapa de testes.....	52
TABELA 27 –	Dados coletados no Caso 2 da quarta etapa de testes.....	53
TABELA 28 –	Dados coletados no Caso 3 da quarta etapa de testes.....	54

TABELA 29 – Dados coletados no Caso 4 da quarta etapa de testes.....	55
TABELA 30 – Dados coletados no Caso 5 da quarta etapa de testes.....	56
TABELA 31 – Dados coletados no Caso 6 da quarta etapa de testes.....	57

LISTA DE SIGLAS

PO – Pesquisa Operacional

PL – Programação Linear

PCC2.5D – Strip Packing Problem (Problema de Carregamento de Contêineres 2.5D)

PCC3D – Knapsack Loading Problem (Problema de Carregamento de Contêineres 3D)

PCB – Bin Packing Problem (Problema de Carregamento de *Bins*)

PCMC – Multi-Container Loading Problem (Problema de Carregamento de Múltiplos Contêineres)

PLIM – Programação Linear Inteira Mista

SUMÁRIO

1	INTRODUÇÃO.....	4
1.1	Justificativa e motivação	5
1.2	Objetivos	6
1.1.1	Objetivo geral.....	6
1.1.2	Objetivos específicos.....	6
1.3	Estrutura textual	7
2	ESTADO DA ARTE.....	8
3	FUNDAMENTAÇÃO TEÓRICA.....	11
3.1	Pesquisa Operacional	11
3.2	Modelagem e Otimização	12
3.2.1	Programação Linear	13
3.2.2	Programação Linear Inteira Mista.....	15
3.3	O problema de carregamento de contêineres.....	16
4	METODOLOGIA	19
4.1	O modelo proposto na literatura.....	19
4.1.1	Contagem de variáveis e restrições	23
4.2	O modelo desenvolvido neste trabalho	25
4.2.1	Contagem de variáveis e restrições	27
4.3	Exemplo de pior caso.....	28
5	EXPERIMENTOS E RESULTADOS.....	29
5.1	O primeiro conjunto de testes	30
5.1.1	Caso 1.....	31
5.1.2	Caso 2.....	32
5.1.3	Caso 3.....	33
5.1.4	Caso 4.....	34
5.1.5	Caso 5.....	35
5.1.6	Caso 6.....	36
5.2	O segundo conjunto de testes	38
5.2.1	Caso 1.....	38
5.2.2	Caso 2.....	39
5.2.3	Caso 3.....	40
5.2.4	Caso 4.....	41

5.2.5	Caso 5.....	42
5.2.6	Caso 6.....	43
5.3	O terceiro conjunto de testes	45
5.3.1	Caso 1.....	45
5.3.2	Caso 2.....	46
5.3.3	Caso 3.....	47
5.3.4	Caso 4.....	48
5.3.5	Caso 23	49
5.3.6	Caso 24	50
5.4	O quarto conjunto de testes	52
5.4.1	Caso 1.....	52
5.4.2	Caso 2.....	53
5.4.3	Caso 3.....	54
5.4.4	Caso 4.....	55
5.4.5	Caso 5.....	56
5.4.6	Caso 6.....	57
6	CONCLUSÃO E TRABALHOS FUTUROS.....	59
6.1	Conclusão	59
6.2	Trabalhos futuros	60
7	REFERÊNCIAS	61

1 INTRODUÇÃO

De acordo com Bianchi e Saldanha (2005), no decorrer dos anos o processo de logística empresarial tem se tornado crucial, ao lado de análises de finanças, de marketing e de produção, no gerenciamento de empresas. Os processos logísticos estão cada vez mais presentes no planejamento empresarial, tendo por objetivo tornar as empresas mais competitivas.

O crescente interesse de grandes empresas em adotar a logística como método de otimização de seus serviços, deve-se, primordialmente, à necessidade de buscar táticas para garantir ao cliente um atendimento rápido e seguro, considerando o custo benefício como quesito determinante neste processo. Para que as companhias consigam aderir estratégias eficazes é fundamental que as mesmas ofereçam produtos e serviços que atendam as necessidades dos clientes sem gerar gastos desnecessários, garantindo que o produto seja colocado no lugar certo, na hora requerida e na quantidade correta. As preocupações com os custos e a satisfação dos clientes são fatores fundamentais para o desenvolvimento e crescimento de uma empresa, como descrito por Bianchi e Saldanha (2005). A respeito do processo de logística Bowersox e Closs citados por Junqueira, Morabito e Yamashita afirmam que:

A eficiência de algumas atividades logísticas, como transporte, armazenagem, manuseio de materiais e embalagem, está diretamente relacionada, entre outros, com o conceito de unitização de carga. Diversos produtos embalados em caixas são normalmente juntados (ou “unitizados”) em unidades maiores, por meio de dispositivos de unitização de carga, como páletes e contêineres. Estas cargas unitizadas apresentam diversas vantagens e em geral reduzem os custos logísticos. Por exemplo, os tempos de carga e descarga e o congestionamento no ponto de destino são reduzidos; o manuseio é facilitado pela simplificação na verificação das mercadorias, em sua entrada, e no rápido posicionamento para a separação de pedidos; as cargas unitizadas reduzem a quantidade de avarias em trânsito, com a utilização de equipamentos especializados para o transporte. (Junqueira, Morabito e Yamashita 2009, pg. 74).

Desta forma, este trabalho busca otimizar o problema de carregamento (ou disposição) de caixas retangulares em dispositivos de unitização de cargas em três dimensões, como contêineres, carrocerias de caminhões, vagões ferroviários ou paletes. Basicamente, tem-se como objetivo determinar o melhor padrão de

disposição das caixas com a finalidade de maximizar o volume (valor) das caixas, contanto que as mesmas não se sobreponham nos dispositivos de unitização.

1.1 Justificativa e motivação

O estudo de técnicas na área de Pesquisa Operacional (PO) proporcionou um grande salto para resolver problemas ambientados em situações reais cuja análise e dimensão eram praticamente inviáveis quando realizadas manualmente. Através de sua emersão sob a forma gráfica, para representar a realidade aprendida em determinado momento, o uso de técnicas de PO simplifica a visualização da amplitude das variáveis envolvidas no problema sem alterar a essência do mesmo.

A motivação para o desenvolvimento de um modelo computacional para o problema de carregamento de contêineres deve-se ao fato de que a resolução manual do mesmo é inviável quando levada em consideração a grandiosidade e complexidade do mesmo, e, além disso, à necessidade de criar uma solução mais eficiente do que a presente na literatura.

Também deve-se levar em consideração que a eficiência da logística de produtos é essencial para o crescimento e continuidade de empresas no mercado de trabalho. Apesar de existirem softwares no mercado que resolvam este problema de maneira ótima dada uma situação em específico, o custo de tais ferramentas é relativamente alto devido à complexidade do mesmo, pode-se tornar inviável sua aquisição para pequenas e médias empresas arcarem com tais despesas (MARTINS et al, 2014). Por isso, o uso de técnicas de PO, as quais propõem modelos para solucionar problemas, é uma alternativa eficiente para a otimização de processos em diversos cenários.

1.2 Objetivos

Os objetivos deste trabalho foram divididos em objetivo geral e objetivos específicos.

1.1.1 Objetivo geral

O objetivo geral e motivador desse trabalho é desenvolver um modelo matemático que resolva o problema do carregamento de contêineres de uma forma mais eficiente que o apresentado na literatura, levando em consideração o tempo de execução e os recursos computacionais utilizados.

O modelo deverá maximizar o valor total da carga respeitando as restrições de não sobreposição das caixas, isto é, não se deve permitir que uma caixa, ao ser posicionada dentro do contêiner, ocupe o mesmo lugar que outra. Também deve-se considerar que as caixas podem ou não ser posicionadas no contêiner dependendo do seu valor e volume.

1.1.2 Objetivos específicos

Para atingir o objetivo geral, este trabalho também possui os seguintes objetivos específicos:

- Fazer um levantamento teórico de modelos previamente propostos que resolvam o problema de carregamento de contêineres;
- Definir um modelo para resolver o problema com o intuito de ser mais eficiente do que o presente na literatura;
- Realizar a implementação dos dois modelos;
- Realizar uma análise sobre o crescimento no número de variáveis e restrições nos dois modelos;
- Submeter os dois modelos aos mesmos dados de entrada com o intuito de comparar os resultados e concluir qual modelo é mais eficiente.

1.3 Estrutura textual

Essa monografia está dividida em 7 capítulos, no qual a pesquisa realizada foi de uma forma geral apresentada neste primeiro capítulo.

No capítulo 2 serão brevemente apresentados alguns artigos já existentes que abordam o problema tratado nesse trabalho, os métodos aplicados e de que forma os autores lidaram com o mesmo.

No capítulo 3 serão abordados os princípios teóricos em relação a Pesquisa Operacional, a Programação Linear, a Programação Linear Inteira Mista e a modelagem de problemas na área da Otimização. Também será apresentado com mais detalhes, o problema de carregamento de contêineres e o que será considerado no desenvolvimento deste trabalho.

No capítulo 4 serão exibidos a modelagem matemática do problema segundo a literatura e a modelagem do problema que foi desenvolvida ao decorrer deste trabalho, com apresentação dos cálculos a respeito da contagem de variáveis e restrições que cada modelo utiliza.

No capítulo 5 serão explicados os critérios que foram levados em consideração na etapa de testes de ambos os modelos, como foram realizados e os resultados obtidos.

No capítulo 6 a monografia é finalizada com as conclusões obtidas após a análise da pesquisa e seus resultados. Sugestões de trabalhos futuros na área são dadas ao final e a bibliografia pesquisada para a realização deste trabalho é apresentada logo na sequência.

2 ESTADO DA ARTE

Uma das primeiras abordagens para o problema de carregamento de contêineres foi apresentada por Gilmore e Gomory (1963, 1965) no qual foi apresentada uma técnica para solucionar o problema utilizando o método Simplex (supondo que ele pode ser modelado como um programa linear) e em um subproblema que deve ser resolvido em cada iteração do simplex para produzir um padrão de carregamento. Desde então, várias abordagens foram propostas para solucionar o problema.

Morabito (1994) examinou alguns modelos existentes de otimização para o problema de corte e cita uma experiência real para a aplicação destes modelos em uma indústria de papel, papelão e móveis. Modificações nos modelos foram realizadas para que restrições de unidades de estoque com tamanhos e custos diferentes e tolerância na demanda de clientes sejam incorporadas.

Bischoff e Ratcliff (1995) argumentaram que as pesquisas existentes na área de carregamento de contêineres eram aplicáveis apenas, a uma restrita parte do espectro de situações encontradas na prática. A fim de solucionar esse problema foram desenvolvidos modelos com considerações de estabilidade, agrupamento de itens e múltiplos destinos.

Chen, Lee e Shen (1995) definiram um modelo matemático de programação inteira mista para resolver o problema de carregamento de contêineres levando em consideração múltiplos contêineres, caixas de tamanhos diferentes, orientação das caixas e não-sobreposição das mesmas.

Scheithauer et al. (1996) exploraram, de maneira heurística, o problema de posicionar caixas em paletes e contêineres de modo a minimizar a quantidade total de dispositivos de unitização de carga utilizados, levando em consideração restrições de empilhamento, estabilidade, múltiplos destinos e limite de peso.

Bram (1996) apresentou uma heurística que utiliza técnicas de busca baseadas na população de caixas para adicionar à resolução do problema restrições que levam em consideração a ordem em que as caixas devem ser retiradas do contêiner. O algoritmo apresentado por este, pode ser usado tanto para duas quanto para três dimensões. Ao invés do posicionamento ser realizado camada por camada, o carregamento é feito caixa por caixa. Segundo o autor, nesse método são adotados critérios locais para se decidir onde as caixas serão alocadas.

Considerando que os critérios não abrangem o contêiner como um todo, pode-se considerar essa estratégia como desvantajosa para a solução do problema.

Terno, Scheithauer e Sommerweiß (1997) apresentaram uma heurística que utiliza algoritmos *branch and bound* para resolver o problema de posicionar um conjunto de caixas diferentes em múltiplos paletes, com a finalidade de minimizar o número de paletes a serem utilizados.

Morabito e Arenales (1997) discutiram um modelo matemático cujo objetivo principal é de posicionar o maior número de caixas a fim de maximizar o volume total ocupado no contêiner. Esse modelo trabalha com caixas de baixa densidade e satisfaz restrições de estabilidade do carregamento. Abordagens conhecidas que utilizam procedimentos de carregamento em duas etapas (carregamento em camadas horizontais e em pilhas verticais) são apresentados.

Ratcliff e Bischoff (1998) propuseram um algoritmo que resolvesse o problema considerando restrições de empilhamento e fragilidade das caixas, visto que, até aquele momento, poucos trabalhos haviam tratado tais restrições.

Terno, Scheithauer e Sommerweiß (2000) apresentaram um método heurístico para resolver o problema de carregamento de contêineres em que o carregamento completo de grupos de itens é levado em consideração.

Bortfeldt e Gehring (2001) desenvolveram um algoritmo genético híbrido com considerações de estabilidade de carga, limite de peso e distribuição de peso dentro do contêiner, aplicável para dispositivos de unitização de cargas tridimensionais.

Pisinger (2002) mostrou uma heurística para resolver o problema do carregamento de um subconjunto de caixas retangulares em um contêiner de dimensões fixas com o objetivo de maximizar o volume total de caixas empacotadas. Para isso, utiliza-se a técnica de decompor o problema em camadas para serem tratadas individualmente.

Silva e Soma (2003) apresentaram um algoritmo polinomial, capaz de encontrar boas soluções para o problema de empacotamento de *bins* tridimensionais. A maior contribuição, foram as considerações sobre a estabilidade estática da carga.

Eley (2003) apresentou um algoritmo que minimiza o número de contêineres utilizados em situações em que o carregamento não pode ser agrupado em um único contêiner. Dentre as restrições utilizadas constam o agrupamento e separação de itens e a distribuição de peso dentro do container.

Junqueira, Morabito e Yamashita (2009) apresentaram modelos matemáticos para a solução do problema levando em consideração restrições de estabilidade e empilhamento de caixas.

Armentano e Araújo (2010) apresentaram uma heurística de solução para o problema considerando restrições de múltiplos destinos. As demandas devem ser entregues, uma de cada vez, de forma a evitar que o contêiner seja parcialmente descarregado e carregado muitas vezes.

Bortfeldt e Wäscher (2013) apresentaram uma ampla revisão de trabalhos publicados de 1981 até 2012, que tratam do problema do carregamento de contêineres. Fatores que precisam ser considerados são identificados (limite e distribuição de peso, prioridade no carregamento, orientação, estabilidade e empilhamento das caixas e agrupamento da carga). Também foi mostrado como esses fatores são representados nos métodos apresentados para a solução deste problema.

O modelo apresentado por Ranck, Yanasse e Morábito (2014) abordou o problema em que caixas de diferentes tipos devem ser carregadas em um contêiner com ou sem camadas horizontais separadas por divisórias. Diferentes destinos e uma rota pré-definida são levados em consideração juntos com a estabilidade do carregamento, resistência ao empilhamento das caixas e balanceamento de peso dentro do contêiner.

3 FUNDAMENTAÇÃO TEÓRICA

Para embasamento teórico no desenvolvimento deste trabalho são apresentados dois conceitos: Pesquisa Operacional e Programação Linear Inteira Mista.

Para que se resolva qualquer problema na área da Otimização, ou mais especificamente, o problema de carregamento de contêineres é necessário que se encontre uma forma de representar essa realidade na forma de um modelo matemático.

Neste capítulo são apresentados de forma breve os aspectos fundamentais teóricos da PO e da PLIM. Por fim são expostos conceitos sobre a modelagem e um detalhamento do problema em questão é feito ao fim.

3.1 Pesquisa Operacional

Silva et al. (1998) definiu a Pesquisa Operacional como a utilização de um método de descrição de um sistema organizado, auxiliado por um modelo, de forma que experimentações do modelo levam à maneira ótima de operar um sistema. De maneira análoga, Andrade (1998) definiu a PO como um método científico para tomar decisões, por meio da elaboração de modelos, que permitem simulações e servem de auxílio para as decisões.

De acordo com Pereira (2009), as raízes da PO se encontram na Segunda Guerra Mundial, quando cientistas foram reunidos pelos comandos de militares britânicos e americanos para desenvolver métodos de alocação de recursos escassos, como radares, submarinos e aviões, para servirem a um grande número de alvos e operações militares (por isso utilizou-se o termo “Operacional”). Com o crescimento econômico após o término da Guerra, esses métodos e ferramentas criados, passaram a ser aplicados nos ramos comercial, governamental e industrial. Porém, os recursos que passaram a ser alocados eram máquinas, matérias-primas, pessoas e relacionados (PEREIRA, 2009).

Como afirmado anteriormente, a utilização de técnicas de pesquisa operacional atraiu o interesse de diversas outras áreas. A natureza dos problemas encontrados é bastante ampla e complexa, o que exige uma abordagem que permita reconhecer os diversos aspectos envolvidos. Uma característica importante da

Pesquisa Operacional é a utilização de modelos. Eles facilitam o processo de análise e de decisão e também permitem a experimentação da solução proposta, isto é, uma decisão pode ser mais bem avaliada e testada antes de ser efetivamente implementada. (LISBOA, 2002)

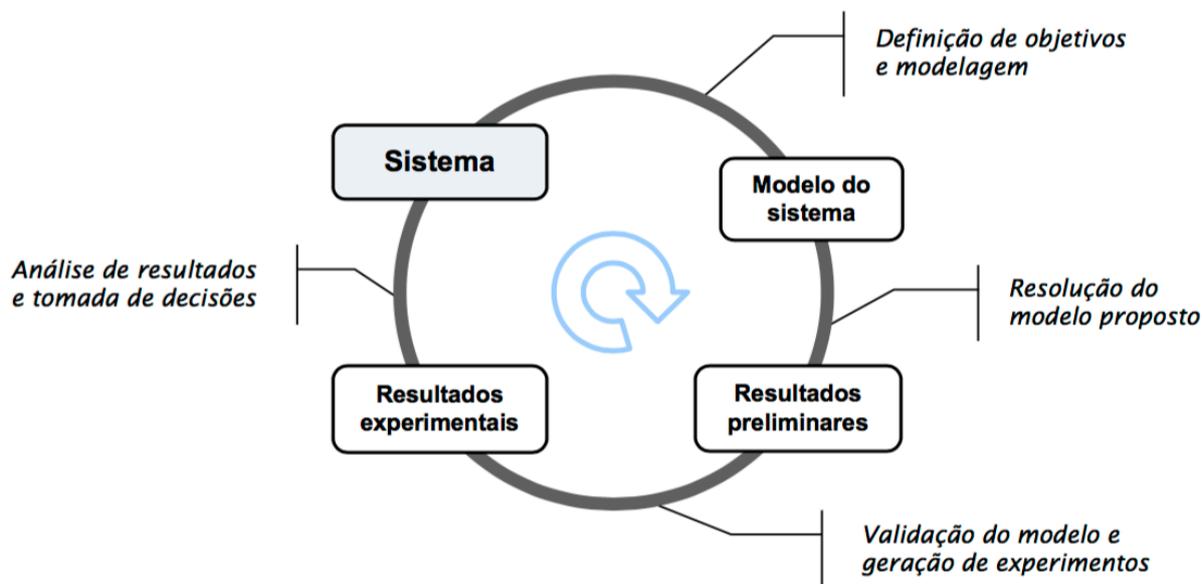
A avaliação de dados sobre atividades em grande volume é permitida devido ao desenvolvimento de técnicas/metodologias juntamente com o suporte e a disseminação de meios computacionais. A multidisciplinaridade da Pesquisa Operacional justifica a extensão dessa disciplina científica em auxílio de quase todos os domínios da atividade humana (TEIXEIRA, 2011).

3.2 Modelagem e Otimização

Para a solução de problemas é necessária a criação de modelos que representem a realidade e, assim, encontrar um resultado para explicar como resolver o problema em questão. Segundo Goldbarg e Luna (2000), um problema pode ser entendido como a dificuldade que impossibilita o cumprimento de uma vontade. Por isso, a solução do problema, que é a arte de criar/escolher modelos que funcionem na prática e sejam eficientes, dão o suporte para sanar as dificuldades do problema através da construção de algoritmos que resolvam tais modelos que representam a realidade de forma simplificada (mantendo proporções equivalentes com a realidade).

Pereira (2009) afirmou que depois que se define o problema a ser resolvido, os analistas de Pesquisa Operacional desenvolvem modelos dos sistemas em questão, para que, com o auxílio deles, possa-se prever e comparar o resultado de alternativas de decisão e estratégias de controle. De forma simplificada, a Figura 1 ilustra as principais fases de um estudo de Pesquisa Operacional.

Figura 1 - Fases de um estudo de Pesquisa Operacional



Fonte: PEREIRA, 2009

Na prática, um projeto de Pesquisa Operacional nem sempre é feito na forma sequencial mostrada na Figura 1. Os resultados preliminares, por exemplo, podem evidenciar inconsistências no modelo, levando a uma redefinição da formulação inicial. Ainda assim, as etapas mostradas na Figura 1 podem ser utilizadas na aplicação de qualquer ferramenta das duas grandes áreas da Pesquisa Operacional: a Otimização e a Simulação.

3.2.1 Programação Linear

A Programação Linear (PL) é uma das técnicas mais poderosas em quase todos os ramos da atividade humana. Entre seus benefícios se destacam, a diminuição dos custos e o aumento dos lucros que são exatamente aqueles procurados por qualquer empresa (PRADO, 2012).

Teixeira (2011) definiu a Programação Linear como uma técnica que utiliza instrumentos matemáticos que permitem a otimização de operações. Ela é largamente utilizada na resolução de problemas que possuem modelos representados por expressões lineares:

Os problemas de Programação Linear referem-se à distribuição eficiente de recursos limitados entre atividades competitivas, com a finalidade de atender a um determinado objetivo, por exemplo, maximização de lucros ou minimização de custos. Em se tratando de programação linear, esse

objetivo será expresso por uma função linear, à qual se dá o nome de função objetivo. É claro que é necessário dizer quais as atividades que consomem cada recurso, e em que proporção é feito esse consumo. Essas informações serão fornecidas por equação ou inequações lineares, uma para cada recurso. Ao conjunto dessas equações ou inequações lineares dá-se o nome de restrição do modelo. Geralmente existem inúmeras maneiras de distribuir os escassos recursos entre as diversas atividades, bastando para isso que essas distribuições sejam coerentes com as equações de consumo de cada recurso, ou seja, que elas satisfaçam as restrições do problema. Entretanto, deseja-se achar aquela distribuição que satisfaça as restrições do problema, e que alcance o objetivo desejado, isto é, que maximize o lucro ou minimize o custo. A essa solução dá-se o nome de solução ótima. Uma vez obtido o modelo linear, constituído pela função objetivo (linear) e pelas restrições lineares, a programação linear se incumbe de achar a sua solução ótima. (TEIXEIRA apud INTRODUÇÃO À PESQUISA OPERACIONAL, 1980).

Segundo Barra (2008), um modelo de Programação Linear como um modelo matemático de otimização no qual todas as funções (tanto a função objetivo, quanto as restrições) são lineares, ou seja, são da forma $f(x) = a \cdot x + b$ na qual a e b são reais e $a \neq 0$. As características que um modelo de um determinado sistema deve apresentar para ser representado por meio de PL são as seguintes:

- Proporcionalidade: a quantidade de recurso consumido por uma dada atividade deve ser proporcional ao nível dessa atividade na solução final do problema. Além disso, o custo de cada atividade é proporcional ao nível de operação da atividade;
- Não Negatividade: deve ser sempre possível desenvolver dada atividade em qualquer nível não negativo e qualquer proporção de um dado recurso deve sempre poder ser utilizado;
- Aditividade: o custo total é a soma das parcelas associadas a cada atividade;
- Separabilidade: pode-se identificar de forma separada o custo (ou consumo de recursos) específico das operações de cada atividade.

De acordo com Garcia et al. (1997), a formulação matemática de um modelo para um problema de Programação Linear deve seguir o esquema abaixo:

Maximizar ou Minimizar:

$$Z = c_1x_1 + c_2x_2 + \dots + c_nx_n \tag{1}$$

Sujeito a:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \quad (2)$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2$$

...

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$$

$$x_i \geq 0 \text{ e } b_j \geq 0 \quad \text{para } i = 1, 2, \dots, n \text{ e } j = 1, 2, \dots, m$$

Em que:

z = função a ser maximizada ou minimizada (geralmente ganho ou custo),
respeitando o conjunto de elementos do problema ou restrições;

x_i = variáveis de decisão que representam as quantidades ou recursos que se quer
determinar para otimizar o resultado global;

c_i = coeficientes de ganho ou custo que cada variável é capaz de gerar;

b_j = quantidade disponível de cada recurso;

a_{ij} = quantidade de recurso que cada variável de decisão consome.

- (1) É a função matemática que modela o critério de qualidade da solução, denominada função objetivo.
- (2) É o conjunto de desigualdades matemáticas que modelam as restrições do problema.

3.2.2 Programação Linear Inteira Mista

No ramo da Pesquisa Operacional existem diversos problemas que podem ser considerados como de Programação Linear Inteira Mista, sendo que estes diferem dos de Programação Linear por possuírem variáveis contínuas e inteiras para satisfazerem um conjunto de restrições lineares de igualdade e desigualdade (MORO, 2000).

Segundo Barboza (2005), a obtenção de uma solução ótima em problemas de otimização linear mista pode ser difícil devido a sua natureza combinatorial. Primeiramente, a forma abordada para resolver o problema seria o uso da

Programação Linear para todas as combinações de variáveis inteiras, e a extração da solução através do menor valor da função objetivo. Entretanto, como o número de combinações cresce exponencialmente com o número de variáveis binárias, essa abordagem se torna inviável para problemas que envolvam um grande número de variáveis inteiras. Outra alternativa seria a relaxação das restrições de integralidade para tratar as variáveis como inteiras e contínuas, porém, ao utilizar essa técnica não se garante a solução ótima do problema.

3.3 O problema de carregamento de contêineres

O problema em questão foi inicialmente estudado por Gilmore e Gomory em (GILMORE; GOMORY, 1961), (GILMORE; GOMORY, 1963) e (GILMORE; GOMORY, 1965) na década de 60, e a partir de então, foram apresentados numerosos documentos e algoritmos para sua solução.

Como proposto anteriormente, o objetivo deste trabalho é desenvolver um modelo matemático eficiente para resolver o problema de carregamento de contêineres. Pisinger (2002) citado por Junqueira, Morabito e Yamashita (2009) divide o problema de carregar caixas dentro de contêineres em quatro variantes: o Problema de Carregamento de Contêineres 2.5D – PCC2.5D (*Strip Packing Problem*), o Problema de Carregamento de Contêineres 3D – PCC3D (*Knapsack Loading Problem*), o Problema de Carregamento de Bins – PCB (*Bin Packing Problem*) e o Problema de Carregamento de Múltiplos Contêineres – PCMC (*Multi-Container Loading Problem*).

No PCC2.5D, o contêiner tem duas dimensões fixas e uma dimensão variável e o problema consiste em decidir como carregar todas as caixas dentro do contêiner de modo a minimizar a dimensão variável necessária. No PCB, todos os contêineres têm as mesmas dimensões e um mesmo custo associado, e o problema consiste em decidir como carregar todas as caixas de modo a minimizar o número de contêineres necessários para o transporte. No PCMC, os contêineres não necessariamente têm as mesmas dimensões e mesmos custos, e o problema neste caso consiste em escolher um subconjunto de contêineres para serem carregados, além de decidir como carregar todas as caixas, de modo a minimizar os custos associados à escolha dos contêineres, de acordo com Junqueira, Morabito e Yamashita (2009).

Para a realização desse trabalho, será utilizado o PCC3D. Nessa variante, o contêiner possui as três dimensões fixas, cada caixa tem um valor associado a si e deve-se escolher um subconjunto de caixas para serem carregadas dentro do contêiner e definir um padrão de arranjo a fim de maximizar o valor total do carregamento.

Bischoff & Ratcliff (1995) citados por Junqueira, Morabito e Yamashita (2009) apresentaram treze possíveis considerações que devem ser analisadas ao se tratar da modelagem do problema de carregamento de contêineres. A Tabela 1 a seguir, lista essas considerações e uma breve descrição de cada uma.

Tabela 1 – Considerações apresentadas por Bischoff e Ratcliff (1995)

	Consideração Prática	Descrição
1	Não-sobreposição das caixas	As caixas, ao serem posicionadas dentro do container, não devem ocupar o mesmo espaço que outras.
2	Orientação	Algumas caixas devem ser carregadas dentro do contêiner com orientação pré-estabelecidas.
3	Empilhamento	Leva em consideração a pressão máxima que uma caixa pode suportar ao ter várias outras posicionadas em cima dela.
4	Manuseio	Se uma caixa precisa ser posicionada em uma determinada posição dentro do contêiner devido ao seu tamanho, peso, etc.
5	Estabilidade	Garantir que caixas que precisam ter faces inferiores, superiores e/ou laterais suportadas por faces de outras caixas.
6	Agrupamento de itens	Caixas do mesmo tipo devem ser posicionadas perto umas das outras.
7	Múltiplos destinos	Caixas a serem entregues em múltiplos destinos devem ser posicionadas perto umas das outras para facilitar o descarregamento.
8	Separação de itens	Necessidade de posicionar caixas distantes umas das outras, caso elas não possam estar em contato.
9	Carregamento completo de grupo de itens	É considerado que um subconjunto de caixas deve ser posicionado inteiramente dentro do contêiner.
10	Prioridades de empacotamento	Caixas que apresentam mercadoria com urgência na entrega devem ter prioridade no empacotamento.
11	Complexidade do padrão de empilhamento	Caixas presentes em padrões de carregamento complexos podem demandar esforços maiores de manuseio, devido, por exemplo, às limitações do equipamento que realiza o carregamento/d Descarregamento da carga.

12	Limite de peso	Garantir que o peso total das caixas não exceda o limite de peso máximo do contêiner.
13	Distribuição de peso dentro do contêiner	Garantir que o centro de gravidade do contêiner esteja localizado próximo ao centro geométrico da base do mesmo.

Fonte: Adaptado de JUNQUEIRA, MORABITO e YAMASHITA, 2009

No desenvolvimento desse trabalho será levado em consideração apenas as restrições de não-sobreposição das caixas na criação do modelo matemático para resolver o problema, ou seja, a consideração 1 apresentada por Bischoff e Ratcliff (1995).

4 METODOLOGIA

Este capítulo descreve os procedimentos metodológicos e os instrumentos utilizados na realização deste trabalho. Cabe salientar que o presente trabalho tem como proposta, desenvolver um modelo matemático que resolva o problema de carregamento de contêineres de uma forma mais eficiente que o apresentado na literatura.

4.1 O modelo proposto na literatura

De acordo com Junqueira, Morabito e Yamashita (2009), seja um dispositivo de unitização de carga qualquer (contêiner, caminhão, vagão ferroviário, etc) de comprimento L , largura W e altura H conhecidos, que deve ser carregado com caixas de diferentes tipos com dimensões l_i, w_i e h_i , quantidade b_i e valor $v_i, i = 1, \dots, n$. Admite-se que as dimensões são inteiras e que as caixas só podem ser empacotadas com seus lados paralelos aos lados do contêiner.

Adotando-se um sistema de coordenadas cartesianas com origem no canto inferior frontal esquerdo do contêiner, seja (p, q, r) a posição onde o canto inferior frontal esquerdo de uma caixa i é posicionado e (s, t, u) a posição onde o canto inferior frontal esquerdo de um caixa j é posicionado. Tem-se que as possíveis posições em que uma caixa posicionada ao longo do comprimento L , largura W e altura H de um contêiner podem ser definidas por:

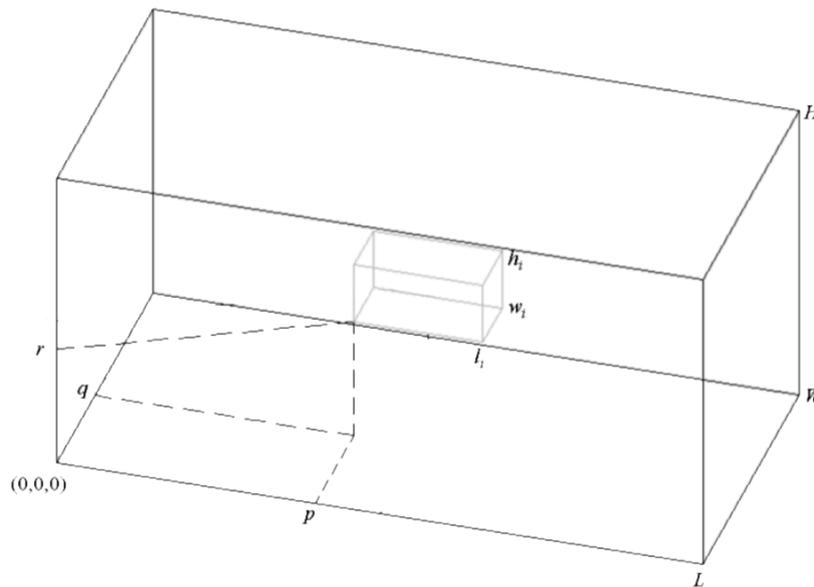
$$X_i = \{ p \mid 0 \leq p \leq L - l_i \text{ e inteiro}, i = 1, \dots, n \} \quad (3)$$

$$Y_i = \{ q \mid 0 \leq q \leq W - w_i \text{ e inteiro}, i = 1, \dots, n \} \quad (4)$$

$$Z_i = \{ r \mid 0 \leq r \leq H - h_i \text{ e inteiro}, i = 1, \dots, n \} \quad (5)$$

A Figura 2 ilustra uma possível disposição de uma caixa do tipo i de dimensões l_i , w_i e h_i dentro de um contêiner de dimensões L , W e H .

Figura 2 – Exemplo de posicionamento de uma caixa do tipo i dentro do contêiner



Fonte: Adaptada de JUNQUEIRA, MORÁBITO e YAMASHITA 2009

De acordo com Junqueira, Morábito e Yamashita (2009), o problema de carregamento de caixas dentro de um único contêiner, considerando apenas a restrição de não-sobreposição de caixas, pode ser formulado como uma extensão direta do modelo de programação linear inteira 0-1 proposto em Beasley (1985) para o problema de corte bidimensional não guilhotinado restrito:

$$\max \sum_{i=1}^n \sum_{p \in X_i} \sum_{q \in Y_i} \sum_{r \in Z_i} v_i \cdot x_{ipqr} \quad (6)$$

$$\sum_{p \in X_i} \sum_{q \in Y_i} \sum_{r \in Z_i} x_{ipqr} \leq 1 \quad (7)$$

$$x_{ipqr} \cdot p - (x_{jstu} \cdot s + l_j) + M \cdot b_{ipqrjstu1} \geq 0 \quad (8)$$

$$y_{ipqr} \cdot q - (y_{jstu} \cdot t + w_j) + M \cdot b_{ipqrjstu2} \geq 0$$

$$z_{ipqr} \cdot r - (z_{jstu} \cdot u + h_j) + M \cdot b_{ipqrjstu3} \geq 0$$

$$x_{jstu} \cdot s - (x_{ipqr} \cdot p + l_i) + M \cdot b_{ipqrjstu4} \geq 0$$

$$y_{jstu} \cdot t - (y_{ipqr} \cdot q + w_i) + M \cdot b_{ipqrjstu5} \geq 0$$

$$z_{jstu} \cdot u - (z_{ipqr} \cdot r + h_i) + M \cdot b_{ipqrjstu6} \geq 0$$

$$b_{ipqrjstu1} + b_{ipqrjstu2} + b_{ipqrjstu3} + b_{ipqrjstu4} + b_{ipqrjstu5} + b_{ipqrjstu6} \leq 5 \quad (9)$$

$$x_{ipqr}, y_{ipqr}, z_{ipqr}, x_{jstu}, y_{jstu}, z_{jstu} \in \{0,1\} \quad \text{Para toda caixa } i \neq j$$

$$b_{ipqrjstun} \in \{0,1\} \quad \text{Para toda caixa } i \neq j$$

M é uma constante muito grande

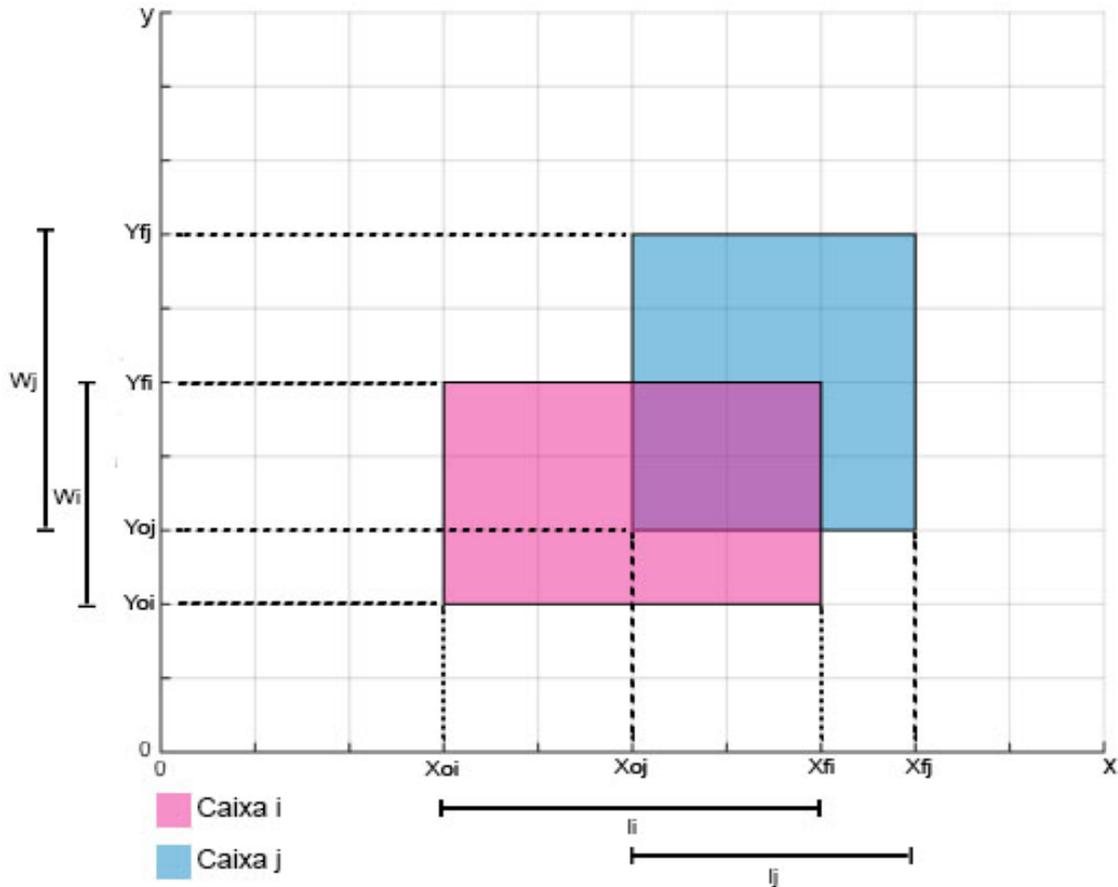
O modelo acima opta por atribuir mais de uma variável binária para cada caixa. Considerando uma caixa i de comprimento l_i , largura w_i e altura h_i , serão atribuídas $(L - l_i + 1) \cdot (W - w_i + 1) \cdot (H - h_i + 1)$ variáveis binárias a ela. Esse valor representa todos os pontos possíveis em que essa caixa pode ser posicionada dentro do contêiner.

Na formulação (6)-(9), a função objetivo (6) visa maximizar o valor total das caixas empacotadas dentro do contêiner, como $v_i = l_i \cdot w_i \cdot h_i$, logo (6) tem o objetivo de maximizar o volume total de caixas a serem inseridas no contêiner.

Como mais de uma variável binária é atribuída a cada caixa, as restrições (7) são responsáveis por garantir que no máximo uma variável relacionada a uma caixa seja diferente de zero, pois o modelo leva em consideração somente o ponto inferior frontal esquerdo no posicionamento de caixas no contêiner.

O conjunto de restrições (8) impedem que haja sobreposição entre as caixas. A Figura 3 ilustra um possível caso de sobreposição entre duas caixas nos eixos xy :

Figura 3 – Sobreposição entre duas caixas nos eixos xy



Fonte: Elaborada pelo autor

O esquema de sobreposição funciona de maneira análoga para os eixos xz e yz .

$$\text{Para a caixa } i: \begin{cases} X_{oi} = X_{ipqr} \cdot p \\ X_{fi} = (X_{ipqr} \cdot p) + l_i \\ Y_{oi} = Y_{ipqr} \cdot q \\ Y_{fi} = (Y_{ipqr} \cdot q) + w_i \\ Z_{oi} = Z_{ipqr} \cdot r \\ Z_{fi} = (Z_{ipqr} \cdot r) + h_i \end{cases}$$

$$\text{Para a caixa } j: \begin{cases} X_{oj} = X_{jstu} \cdot s \\ X_{fj} = (X_{jstu} \cdot s) + l_j \\ Y_{oj} = Y_{jstu} \cdot t \\ Y_{fj} = (Y_{jstu} \cdot t) + w_j \\ Z_{oj} = Z_{jstu} \cdot u \\ Z_{fj} = (Z_{jstu} \cdot u) + h_j \end{cases}$$

No qual (p, q, r) é o ponto em que será posicionada a caixa i e (s, t, u) o ponto em que será posicionada a caixa j . As três primeiras inequações do conjunto de restrições (8) analisam se a caixa j está à esquerda da caixa i em relação aos eixos x, y e z respectivamente. Enquanto as três última inequações analisam se a caixa i está à esquerda da caixa j em relação aos eixos x, y e z . Assim garantindo a não sobreposição para qualquer tipo de situação.

M é uma constante muito grande que é responsável por garantir a veracidade de cada inequação do conjunto de restrições (8). Se $b_{ipqrjstu}$ for igual a 1 existe sobreposição entre as caixas, se for igual a 0 não existe sobreposição.

O conjunto de restrições (9) conta, para cada par de caixas i, j , o número de lados sem colisão. De modo que, considerando o problema tridimensional, a colisão não deve ocorrer em pelo menos um dos eixos (x, y ou z) para que as caixas i, j não estejam sobrepostas.

4.1.1 Contagem de variáveis e restrições

Ao se somar o número de variáveis total do tipo x_{ipqr} ao número total de variáveis do tipo $b_{ipqrjstuk}$ obtem-se o número total de variáveis do sistema. Que também pode ser encontrado através do seguinte cálculo:

$$\left[\sum_{i=1}^{n-1} \sum_{j=2}^n [(L - l_i + 1) \cdot (W - w_i + 1) \cdot (H - h_i + 1)] \cdot [(L - l_j + 1) \cdot (W - w_j + 1) \cdot (H - h_j + 1)] \right] \cdot 7 \quad (10)$$

Como explicado anteriormente, as restrições do tipo (7) são responsáveis por garantir que apenas uma variável por caixa seja diferente de zero, logo, o número total de restrições deste tipo é igual ao número total de caixas do sistema.

Já o grupo de restrições (8), por ser responsável por comparar todas as variáveis de uma caixa com todas as variáveis das outras caixas separadamente em grupos de dois, possui um número total de restrições igual a:

$$\left[\sum_{i=1}^{n-1} \sum_{j=2}^n [(L - l_i + 1) \cdot (W - w_i + 1) \cdot (H - h_i + 1)] \cdot [(L - l_j + 1) \cdot (W - w_j + 1) \cdot (H - h_j + 1)] \right] \cdot 6 \quad (11)$$

Para toda caixa $i \neq j$

- i. Em que n equivale ao número total de caixas;
- ii. $(L - l_i + 1) \cdot (W - w_i + 1) \cdot (H - h_i + 1)$ representa o número total de variável da caixa i ;
- iii. $(L - l_j + 1) \cdot (W - w_j + 1) \cdot (H - h_j + 1)$ representa o número total de variáveis da caixa j .

O número de restrições do tipo (9) pode ser obtido através da seguinte fórmula:

$$\sum_{i=1}^{n-1} \sum_{j=2}^n [(L - l_i + 1) \cdot (W - w_i + 1) \cdot (H - h_i + 1)] \cdot [(L - l_j + 1) \cdot (W - w_j + 1) \cdot (H - h_j + 1)] \quad (12)$$

Para toda caixa $i \neq j$

Portanto conclui-se que o número total de restrições do modelo é equivalente a soma das que foram apresentadas, conforme é mostrado abaixo:

$$\left[\sum_{i=1}^{n-1} \sum_{j=2}^n [(L - l_i + 1) \cdot (W - w_i + 1) \cdot (H - h_i + 1)] \cdot [(L - l_j + 1) \cdot (W - w_j + 1) \cdot (H - h_j + 1)] \right] \cdot 7 + n \quad (13)$$

4.2 O modelo desenvolvido neste trabalho

Seja um contêiner de comprimento L , largura W e altura H , que deve ser carregado com caixas de diferentes tipos. Cada tipo de caixa possui comprimento l_i , largura w_i , altura h_i e volume v_i , $i = 1, \dots, n$. Admite-se que as dimensões das caixas são reais e que elas só podem ser empacotadas com os seus lados paralelos aos lados do contêiner.

Adotando-se um sistema de coordenadas cartesianas com origem no canto inferior frontal esquerdo do contêiner, seja (x_i, y_i, z_i) , pertencente ao espaço tridimensional, a posição onde o canto inferior frontal esquerdo de uma determinada caixa é colocado. As possíveis posições, ao longo do comprimento L , largura W e altura H do contêiner, onde cada caixa pode ser colocada, podem ser definidas por meio dos conjuntos:

$$X_i = \{x_i \mid 0 \leq x_i \leq L - l_i \text{ e real}, i = 1, \dots, n\} \quad (14)$$

$$Y_i = \{y_i \mid 0 \leq y_i \leq W - w_i \text{ e real}, i = 1, \dots, n\} \quad (15)$$

$$Z_i = \{z_i \mid 0 \leq z_i \leq H - h_i \text{ e real}, i = 1, \dots, n\} \quad (16)$$

O ponto $P(x_i, y_i, z_i)$ diz respeito ao canto inferior frontal esquerdo da caixa que será usada como base para o posicionamento da mesma no contêiner. Tem-se o modelo matemático:

$$\max: \sum_{i=1}^n v_i \cdot d_i \quad (17)$$

Sujeito a:

$$x_i - x_j + M \cdot b_{ij1} \leq -l_i + M \quad (18)$$

$$x_j - x_i + M \cdot b_{ij2} \leq -l_j + M$$

$$y_i - y_j + M \cdot b_{ij3} \leq -w_i + M$$

$$y_j - y_i + M \cdot b_{ij4} \leq -w_j + M$$

$$z_i - z_j + M \cdot b_{ij5} \leq -h_i + M$$

$$z_j - z_i + M \cdot b_{ij6} \leq -h_j + M$$

$$b_{ij1} + b_{ij2} + b_{ij3} + b_{ij4} + b_{ij5} + b_{ij6} - c_{ij} = 0 \quad (19)$$

$$d_i + d_j \leq c_{ij} + 1 \quad (20)$$

$$c_{ij} \geq 0 \quad \text{Para toda caixa } i \neq j$$

$$x_i \geq 0, \quad y_i \geq 0, \quad z_i \geq 0, \quad d_i \in \{0,1\} \quad \text{Para toda caixa } i = 1, 2, \dots, n$$

$$b_{ijk} \in \{0,1\} \quad \text{Para toda caixa } i \neq j$$

M é uma constante muito grande

A função objetivo (17) visa maximizar o valor total das caixas empacotadas dentro do contêiner. Diferente do modelo anterior, apenas uma variável binária d_i é atribuída por caixa nesse modelo. Ela é responsável por dizer se a caixa será ou não inserida dentro do contêiner.

As restrições do tipo (18-20) dizem respeito à não sobreposição das caixas. No qual as restrições (18) dizem respectivamente que:

- i. i está à esquerda de j em relação ao eixo x se $b_{ij1} = 1$;
- ii. i está à direita de j em relação ao eixo x se $b_{ij2} = 1$;
- iii. i está à esquerda de j em relação ao eixo y se $b_{ij3} = 1$;
- iv. i está à direita de j em relação ao eixo y se $b_{ij4} = 1$;
- v. i está abaixo de j em relação ao eixo z se $b_{ij5} = 1$;
- vi. i está acima de j em relação ao eixo z se $b_{ij6} = 1$;

M é uma constante muito grande que é responsável por garantir a veracidade de cada inequação do conjunto de restrições (18). Se b_{ij} for igual a 1 não existe sobreposição entre as caixas, se for igual a 0 existe sobreposição.

As restrições do tipo (19) visam armazenar em c_{ij} a soma das variáveis do tipo b_{ij} , dessa forma, contando, para cada par de caixas i, j , o número de lados

sem colisão.

As restrições do tipo (20) decidem se as caixas d_i e d_j serão posicionadas no contêiner em caso de sobreposição por falta de espaço. Forçando a saída de pelo menos uma caixa em caso de sobreposição. c_{ij} é uma variável real que pode assumir valores entre 0 e 3:

- 0: As caixas i e j estão sobrepostas em todos os três eixos e pelo menos uma delas terá de ser retirada;
- 1: As caixas i e j estão sobrepostas com relação a dois eixos, porém ambas poderão ser posicionadas;
- 2: As caixas i e j estão sobrepostas com relação a um eixo, porém ambas poderão ser posicionadas.
- 3: As caixas i e j não estão sobrepostas com relação a nenhum dos eixos e ambas poderão ser posicionadas.

4.2.1 Contagem de variáveis e restrições

O número de variáveis total do sistema pode ser obtido através da seguinte fórmula:

$$3n + \frac{n^2 - n}{2} \cdot 6 + n + \frac{n^2 - n}{2} = \frac{7n^2 + n}{2} \quad (21)$$

- i. $3n$: pois temos três coordenadas (x_i, y_i, z_i) para cada caixa do tipo i ;
- ii. $\frac{n^2 - n}{2} \cdot 6$: é o número total de variáveis do tipo b_{ijk} ;
- iii. n : é o número total de variáveis do tipo d_i ;
- iv. $\frac{n^2 - n}{2}$: é o número total de variáveis do tipo c_{ij} .

Levando em consideração que as restrições do tipo (18) são responsáveis por comparar todas as caixas em grupos de dois, o número de restrições deste tipo é:

$$\frac{n^2 - n}{2} \cdot 6 \quad (22)$$

O número de restrições do tipo (19) é igual ao número de restrições do tipo (20):

$$\frac{n^2 - n}{2} \quad (23)$$

Logo, o número total de restrições do modelo pode ser obtido a partir da seguinte fórmula:

$$\frac{6(n^2 - n)}{2} + \frac{2(n^2 - n)}{2} = 4n^2 - 4n \quad (24)$$

4.3 Exemplo de pior caso

Considere um contêiner cúbico de lado 100 no qual devem ser posicionadas 100 caixas de lados iguais a 1. De acordo com as fórmulas apresentadas nas seções anteriores, a Tabela 2 apresenta o número total de variáveis e restrições que cada um dos modelos utilizaria para resolver esse problema:

Tabela 2 – Número total de variáveis e restrições que cada modelo utiliza no pior caso

	Modelo 1	Modelo 2
Número de variáveis	$3,2622 \times 10^{16}$	35.050
Número de restrições	$3,2622 \times 10^{16}$	39.600

A discrepância dá-se ao fato de que tanto o número de variáveis quanto o de restrições do modelo 1 dependem inteiramente dos lados do contêiner e das caixas. Logo, quanto maior a diferença entre os lados do contêiner e os lados das caixas maior será o número de restrições e variáveis utilizadas pelo modelo 1. Tal dependência não existe no modelo 2.

5 EXPERIMENTOS E RESULTADOS

O modelo encontrado na literatura e o modelo elaborado neste trabalho foram implementados para serem executados no software MATLAB (versão R2014b 8.4.0.150421) que foi utilizado como interface para a geração das matrizes referentes às restrições dos modelos. As resoluções destes foram realizadas utilizando o recurso solucionador Gurobi (versão 6.5, com parâmetros *default*). Os testes foram feitos em um microcomputador equipado Intel Core i7 2.20 GHz (terceira geração) e 8 GB de memória RAM.

Os modelos foram testados com exemplos gerados de forma não-aleatória a fim de observar o padrão de respostas nas análises comparatórias. Foi estipulado um limite de 3600 segundos para cada teste. Foram observados o tempo de execução e o valor final da função objetivo ao final de cada teste para cada modelo, e a razão entre tais valores foi calculada. Ao final, esses dados foram utilizados para a geração de gráficos a fim de tornarem mais claras as comparações entre os modelos.

Foram escolhidos oito tipos de caixas diferentes com os valores de largura, comprimento e altura variando de 2 a 5. A Tabela 3 mostra os tipos de caixas e suas respectivas dimensões.

Tabela 3 - Tipos de caixas e suas respectivas dimensões

Tipo caixas	Comprimento	Largura	Altura
1	2	2	2
2	3	2	2
3	3	3	2
4	3	3	3
5	3	3	4
6	3	4	4
7	4	4	4
8	4	4	5

Fonte: Elaborada pelo autor

Os testes foram divididos em quatro partes, no qual todas possuem seis casos diferentes. Nos dois primeiros conjuntos é considerado que o volume total das caixas pode ser menor que o volume do contêiner e nos dois últimos o volume total das caixas deve obrigatoriamente ser maior que o volume total do contêinere

(para que nem todas caixas sejam posicionadas).

As quantidades de tipos de caixa e as dimensões do contêiner variam de caso a caso. Optou-se por utilizar um contêiner de forma cúbica, ou seja, $L = W = H$ para facilitar a análise dos resultados.

5.1 O primeiro conjunto de testes

Para a primeira etapa de testes, as dimensões do contêiner foram variadas de 5 a 12. O número de tipos de caixas por caso é representado na Tabela 4.

Tabela 4 – Quantidade de tipos de caixa relacionada a cada caso da primeira etapa de testes

Caso	Quantidade de tipos de caixas
1	2
2	3
3	4
4	5
5	6
6	7

Fonte: Elaborada pelo autor

Observou-se o tempo de execução e o valor final da função objetivo de cada modelo executado e ao final foi calculada a razão entre os valores obtidos pelo o modelo 2 e os valores obtidos pelo modelo 1. Os dados coletados são encontrados nas tabelas seguintes.

5.1.1 Caso 1

Teste realizado considerando que existem duas caixas de cada tipo.

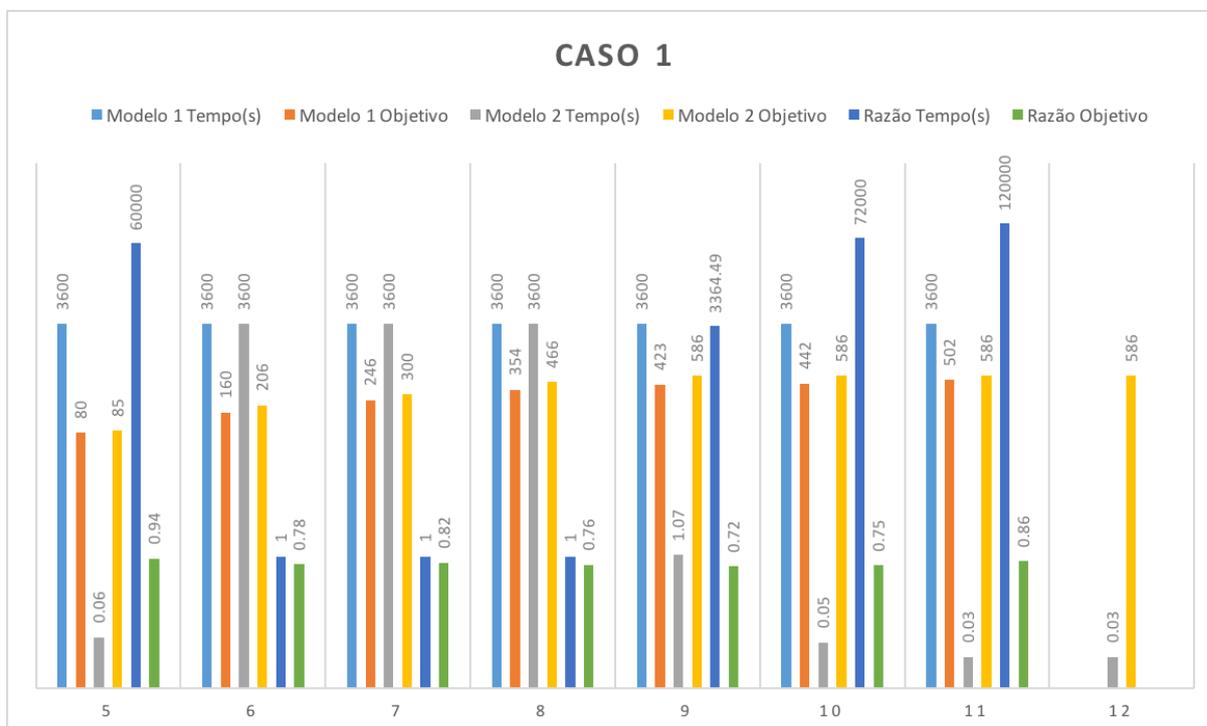
Tabela 5 – Dados coletados no Caso 1 da primeira etapa de testes

Lado do Contêiner	Modelo 1		Modelo 2		Razão	
	Tempo(s)	Objetivo	Tempo(s)	Objetivo	Tempo(s)	Objetivo
5	3600	80	0.06	85	60000	0.94
6	3600	160	3600	206	1.00	0.78
7	3600	246	3600	300	1.00	0.82
8	3600	354	3600	466	1.00	0.76
9	3600	423	1.07	586	3364.49	0.72
10	3600	442	0.05	586	72000	0.75
11	3600	502	0.03	586	120000	0.86
12	*	*	0.03	586	0.00	0.00

* Estouro de memória

Fonte: Elaborada pelo autor

Figura 4 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 1 da primeira etapa de testes



Fonte: Elaborada pelo autor

5.1.2 Caso 2

Teste realizado considerando que existem três caixas de cada tipo.

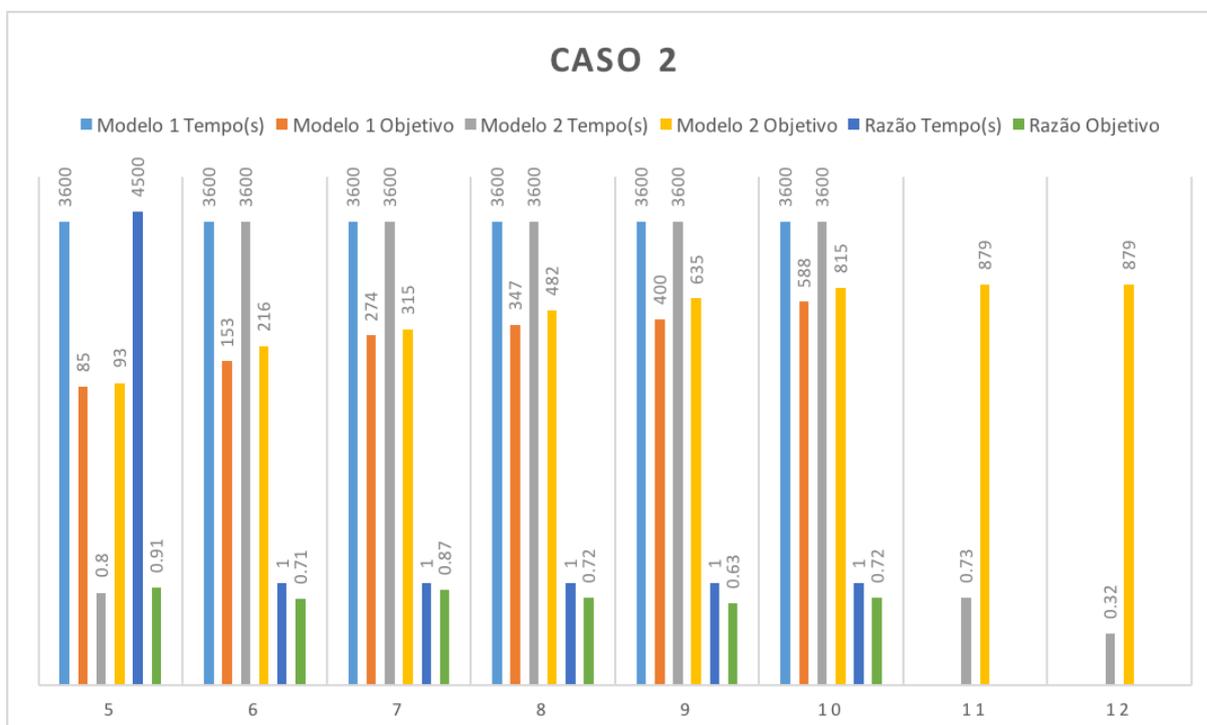
Tabela 6 – Dados coletados no Caso 2 da primeira etapa de testes

Lado do Contêiner	Modelo 1		Modelo 2		Razão	
	Tempo(s)	Objetivo	Tempo(s)	Objetivo	Tempo(s)	Objetivo
5	3600	85	0.8	93	4500	0.91
6	3600	153	3600	216	1	0.71
7	3600	274	3600	315	1	0.87
8	3600	347	3600	482	1	0.72
9	3600	400	3600	635	1	0.63
10	3600	588	3600	815	1	0.72
11	*	*	0.73	879	0	0.00
12	*	*	0.32	879	0	0.00

* Estouro de memória

Fonte: Elaborada pelo autor

Figura 5 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 2 da primeira etapa de testes



Fonte: Elaborada pelo autor

5.1.3 Caso 3

Teste realizado considerando que existem quatro caixas de cada tipo.

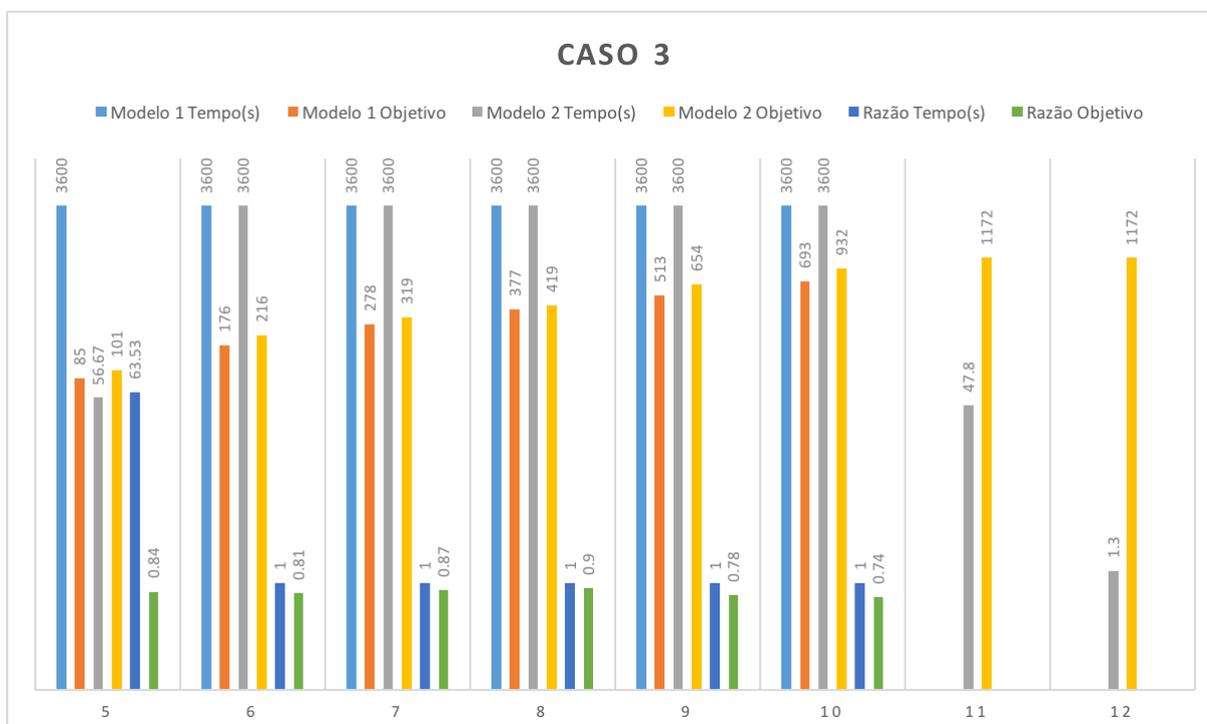
Tabela 7 – Dados coletados no Caso 3 da primeira etapa de testes

Lado do Contêiner	Modelo 1		Modelo 2		Razão	
	Tempo(s)	Objetivo	Tempo(s)	Objetivo	Tempo(s)	Objetivo
5	3600	85	56.67	101	63.53	0.84
6	3600	176	3600	216	1.00	0.81
7	3600	278	3600	319	1.00	0.87
8	3600	377	3600	419	1.00	0.90
9	3600	513	3600	654	1.00	0.78
10	3600	693	3600	932	1.00	0.74
11	*	*	47.8	1172	0.00	0.00
12	*	*	1.30	1172	0.00	0.00

* Estouro de memória

Fonte: Elaborada pelo autor

Figura 6 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 3 da primeira etapa de testes



Fonte: Elaborada pelo autor

5.1.4 Caso 4

Teste realizado considerando que existem cinco caixas de cada tipo.

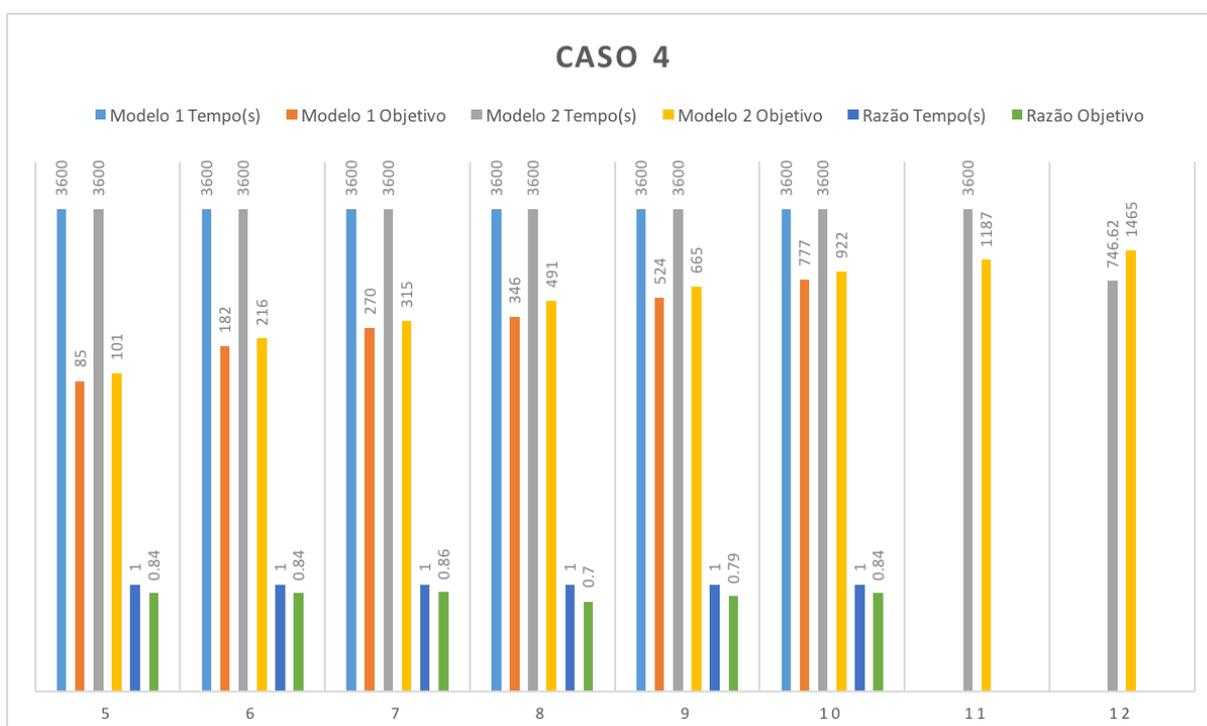
Tabela 8 – Dados coletados no Caso 4 da primeira etapa de testes

Lado do Contêiner	Modelo 1		Modelo 2		Razão	
	Tempo(s)	Objetivo	Tempo(s)	Objetivo	Tempo(s)	Objetivo
5	3600	85	3600	101	1	0.84
6	3600	182	3600	216	1	0.84
7	3600	270	3600	315	1	0.86
8	3600	346	3600	491	1	0.70
9	3600	524	3600	665	1	0.79
10	3600	777	3600	922	1	0.84
11	*	*	3600	1187	0	0.00
12	*	*	746.62	1465	0	0.00

* Estouro de memória

Fonte: Elaborada pelo autor

Figura 7 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 4 da primeira etapa de testes



Fonte: Elaborada pelo autor

5.1.5 Caso 5

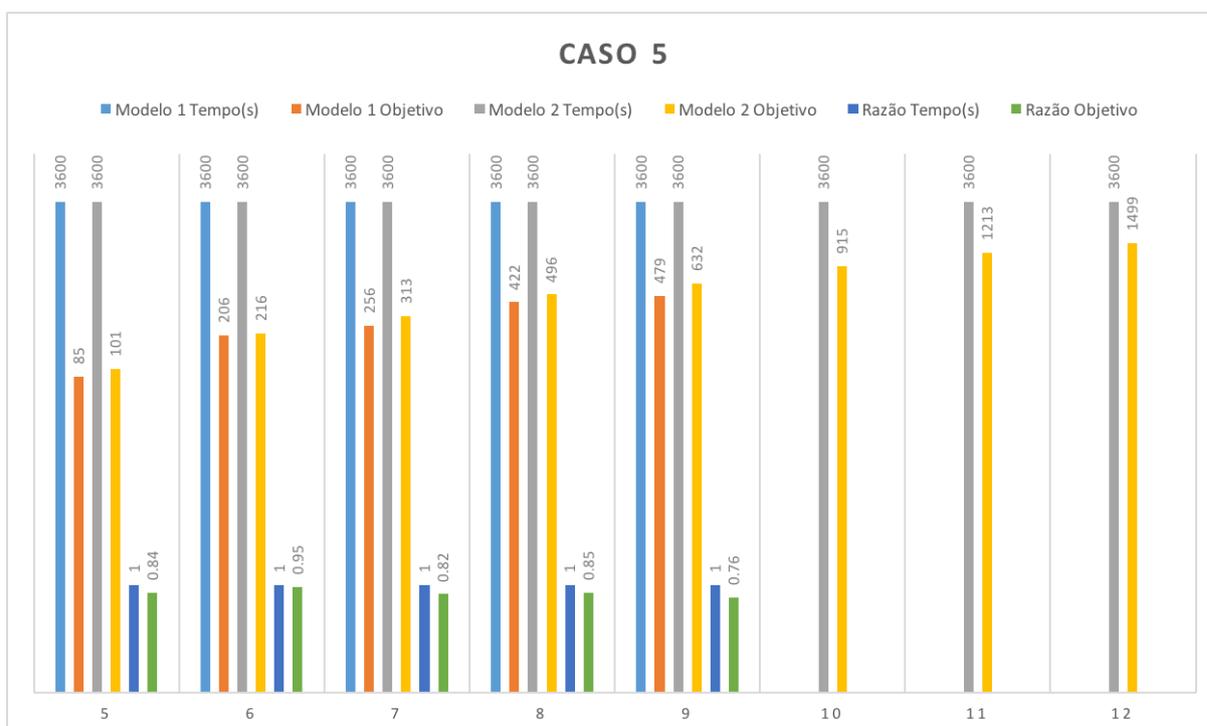
Teste realizado considerando que existem seis caixas de cada tipo.

Tabela 9 – Dados coletados no Caso 5 da primeira etapa de testes

Lado do Contêiner	Modelo 1		Modelo 2		Razão	
	Tempo(s)	Objetivo	Tempo(s)	Objetivo	Tempo(s)	Objetivo
5	3600	85	3600	101	1	0.84
6	3600	206	3600	216	1	0.95
7	3600	256	3600	313	1	0.82
8	3600	422	3600	496	1	0.85
9	3600	479	3600	632	1	0.76
10	*	*	3600	915	0	0.00
11	*	*	3600	1213	0	0.00
12	*	*	3600	1499	0	0.00

* Estouro de memória
Fonte: Elaborada pelo autor

Figura 8 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 5 da primeira etapa de testes



Fonte: Elaborada pelo autor

5.1.6 Caso 6

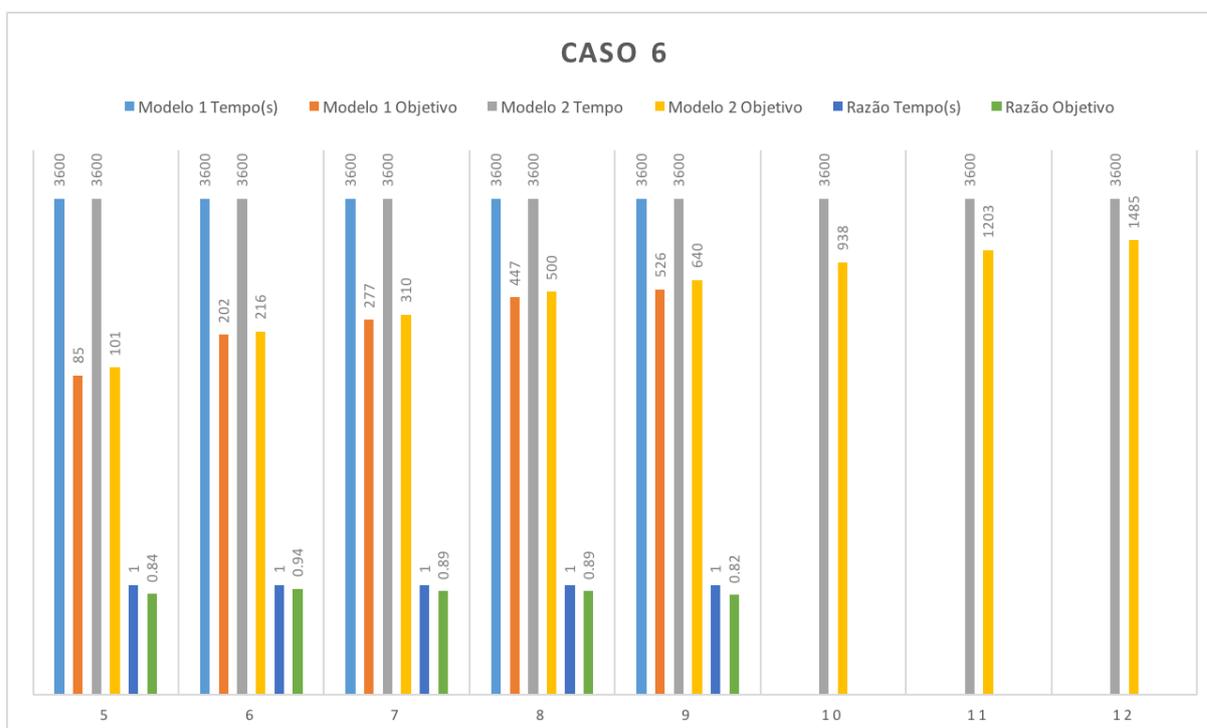
Teste realizado considerando que existem sete caixas de cada tipo.

Tabela 10 – Dados coletados no Caso 6 da primeira etapa de testes

Lado do Contêiner	Modelo 1		Modelo 2		Razão	
	Tempo(s)	Objetivo	Tempo	Objetivo	Tempo(s)	Objetivo
5	3600	85	3600	101	1	0.84
6	3600	202	3600	216	1	0.94
7	3600	277	3600	310	1	0.89
8	3600	447	3600	500	1	0.89
9	3600	526	3600	640	1	0.82
10	*	*	3600	938	0	0.00
11	*	*	3600	1203	0	0.00
12	*	*	3600	1485	0	0.00

* Estouro de memória
Fonte: Elaborada pelo autor

Figura 9 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 6 da primeira etapa de testes



Fonte: Elaborada pelo autor

Através da análise das tabelas e dos gráficos anteriores pode-se observar que o Modelo 2 foi, em todos os testes, mais eficiente do que o Modelo 1 tanto no quesito de tempo de execução e valor final da função objetivo, mesmo que em alguns casos ambos os modelos tenham usado o tempo máximo de 3600 segundos como critério de parada.

Vale a pena ressaltar que o Modelo 1 excedeu a memória do computador que estava sendo utilizado para os testes na parte final de cada um dos casos e que devido a isso os valores de tempo de execução e os valores finais da função objetivo foram anulados.

5.2 O segundo conjunto de testes

A segunda etapa de testes também foi dividida em seis casos diferentes. As dimensões das caixas foram mantidas e os lados do contêiner variaram de 15 a 40 em intervalos de 5. A Tabela 11 mostra o número de tipos de caixas por caso nesta série de testes.

Tabela 11 – Quantidade de tipos de caixa relacionada a cada caso da segunda etapa de testes

Caso	Quantidade de tipos de caixas
1	8
2	9
3	10
4	11
5	12
6	13

Fonte: Elaborada pelo autor

Observou-se o tempo de execução e o valor final da função objetivo de cada modelo executado e ao final foi calculada a razão entre os valores obtidos pelo Modelo 2 e os valores obtidos pelo Modelo 1. Os dados coletados são encontrados nas tabelas seguintes.

5.2.1 Caso 1

Teste realizado considerando que existem oito caixas de cada tipo.

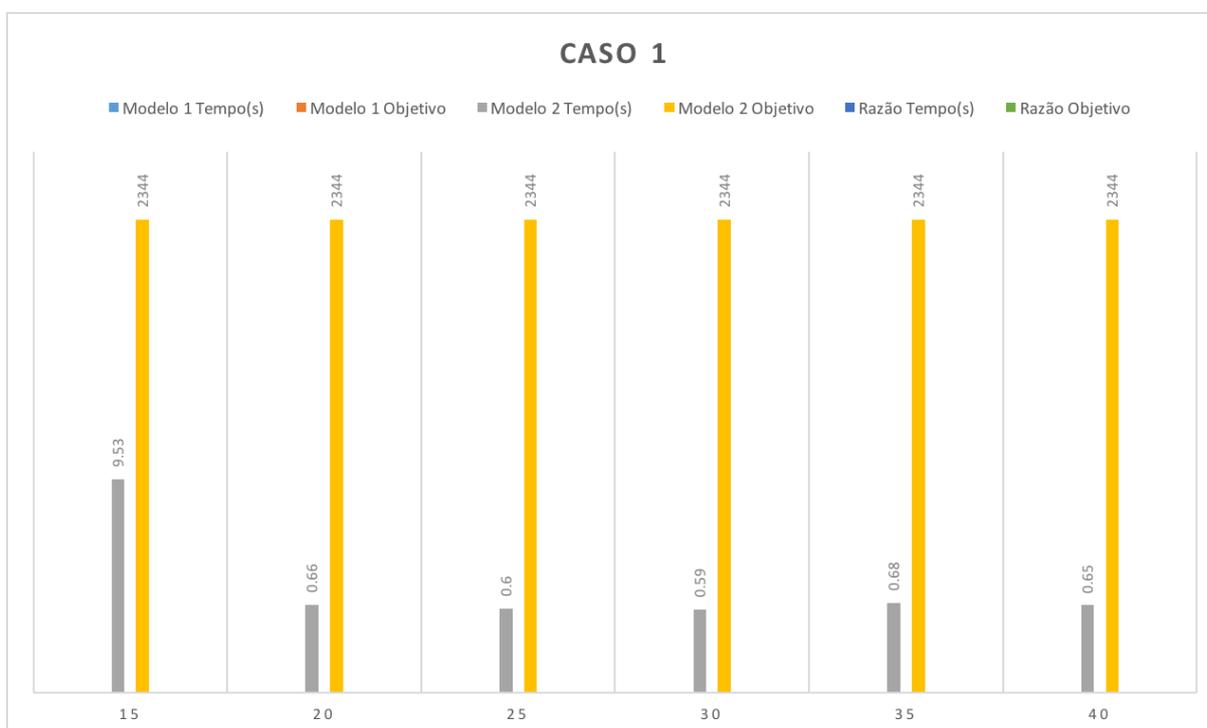
Tabela 12 – Dados coletados no Caso 1 da segunda etapa de testes

Lado do Contêiner	Modelo 1		Modelo 2		Razão	
	Tempo(s)	Objetivo	Tempo(s)	Objetivo	Tempo(s)	Objetivo
15	*	*	9.53	2344	0	0
20	*	*	0.66	2344	0	0
25	*	*	0.60	2344	0	0
30	*	*	0.59	2344	0	0
35	*	*	0.68	2344	0	0
40	*	*	0.65	2344	0	0

* Estouro de memória

Fonte: Elaborada pelo autor

Figura 10 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 1 da segunda etapa de testes



Fonte: Elaborada pelo autor

5.2.2 Caso 2

Teste realizado considerando que existem nove caixas de cada tipo.

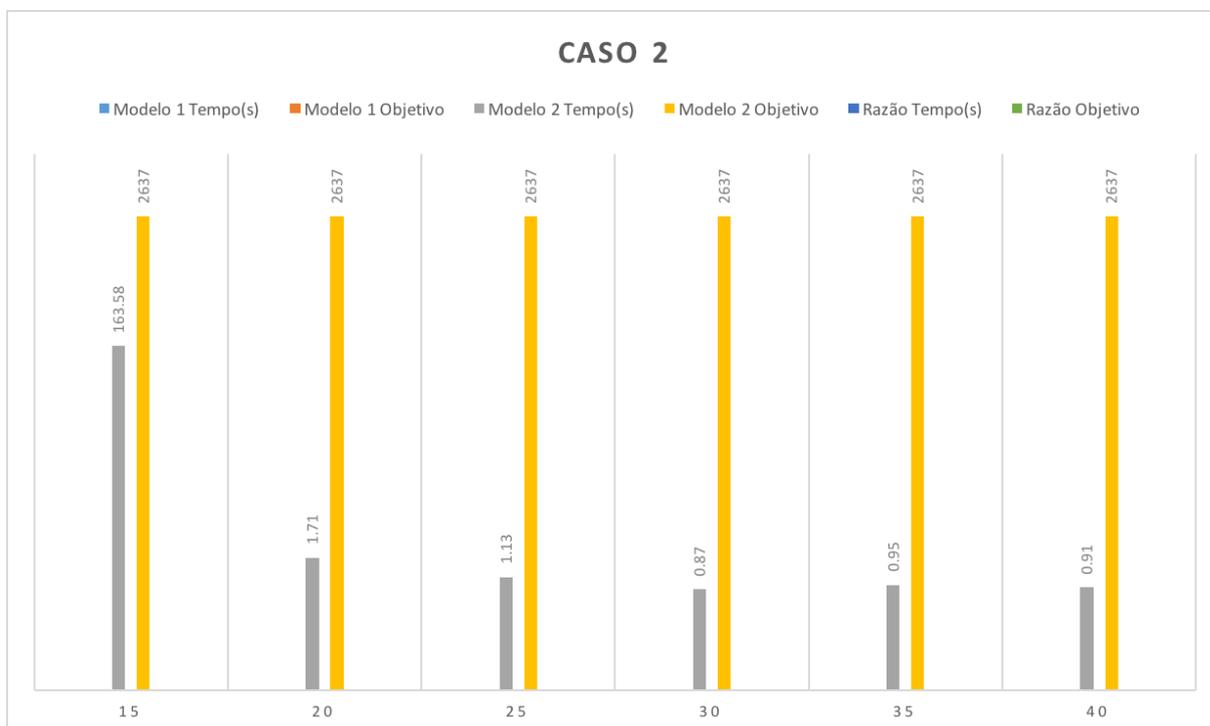
Tabela 13 – Dados coletados no Caso 2 da segunda etapa de testes

Lado do Contêiner	Modelo 1		Modelo 2		Razão	
	Tempo(s)	Objetivo	Tempo(s)	Objetivo	Tempo(s)	Objetivo
15	*	*	163.58	2637	0	0
20	*	*	1.71	2637	0	0
25	*	*	1.13	2637	0	0
30	*	*	0.87	2637	0	0
35	*	*	0.95	2637	0	0
40	*	*	0.91	2637	0	0

* Estouro de memória

Fonte: Elaborada pelo autor

Figura 11 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 2 da segunda etapa de testes



Fonte: Elaborada pelo autor

5.2.3 Caso 3

Teste realizado considerando que existem dez caixas de cada tipo.

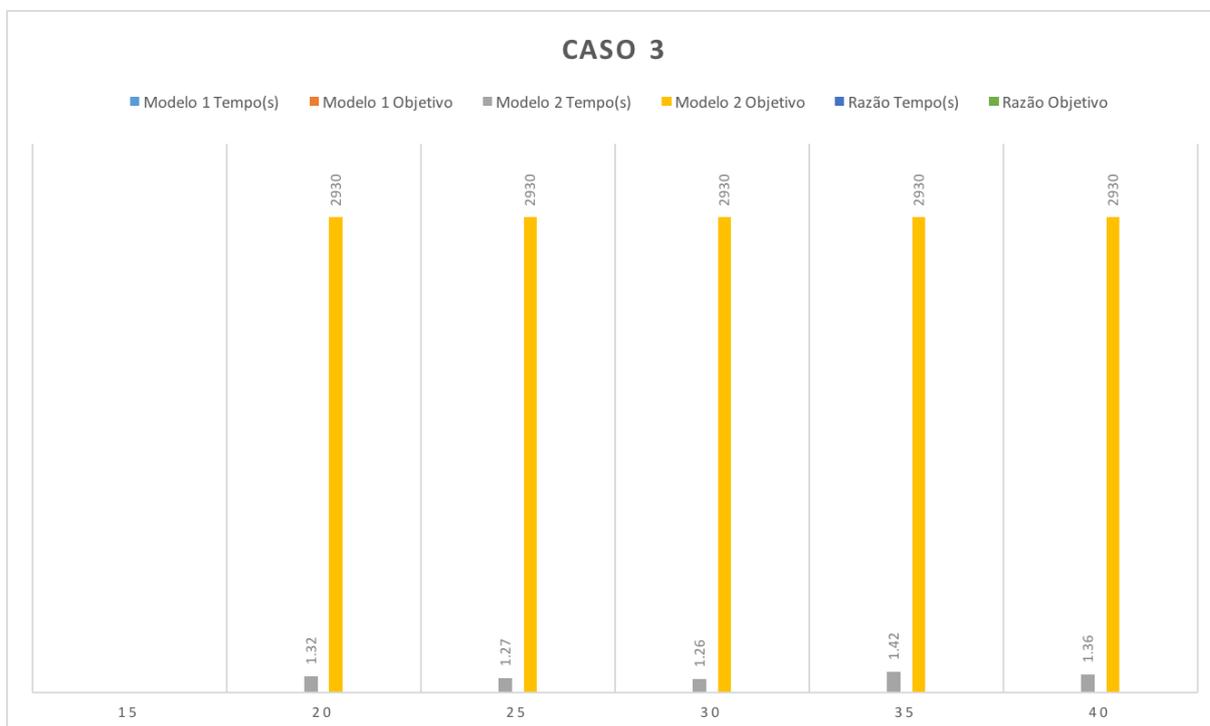
Tabela 14 – Dados coletados no Caso 3 da segunda etapa de testes

Lado do Contêiner	Modelo 1		Modelo 2		Razão	
	Tempo(s)	Objetivo	Tempo(s)	Objetivo	Tempo(s)	Objetivo
15	*	*	3600	2702	0	0
20	*	*	1.32	2930	0	0
25	*	*	1.27	2930	0	0
30	*	*	1.26	2930	0	0
35	*	*	1.42	2930	0	0
40	*	*	1.36	2930	0	0

* Estouro de memória

Fonte: Elaborada pelo autor

Figura 12 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 3 da segunda etapa de testes



Fonte: Elaborada pelo autor

5.2.4 Caso 4

Teste realizado considerando que existem onze caixas de cada tipo.

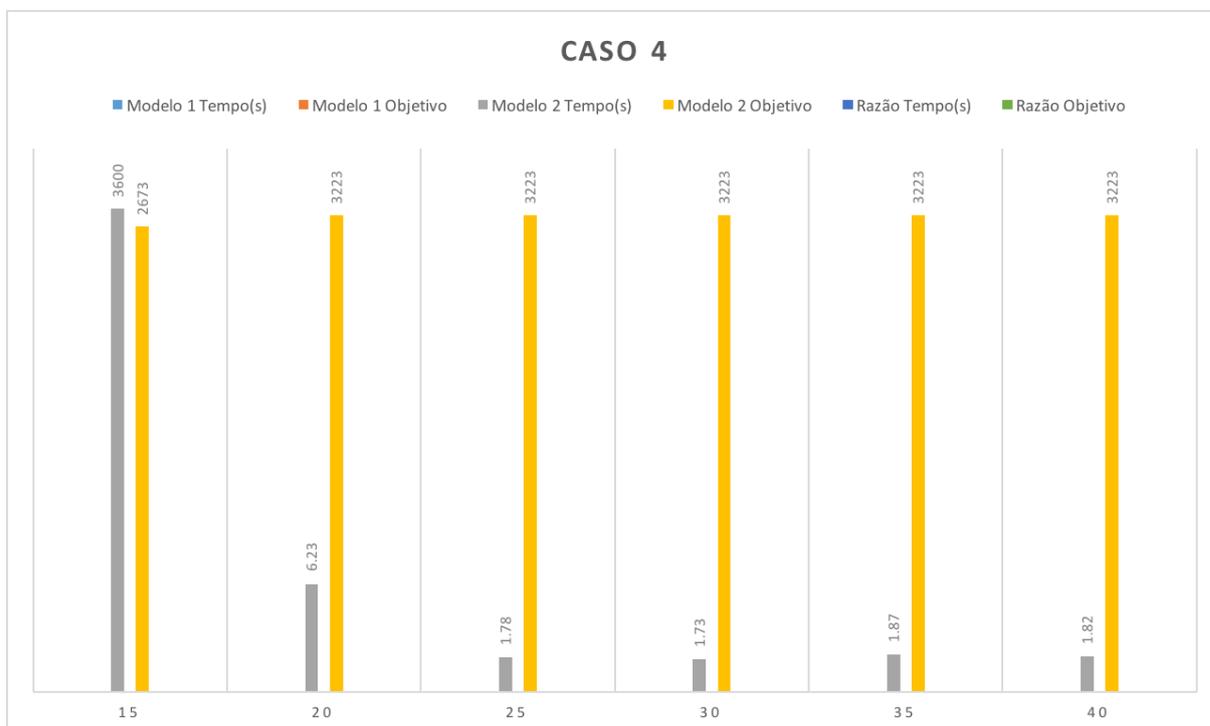
Tabela 15 – Dados coletados no Caso 4 da segunda etapa de testes

Lado do Contêiner	Modelo 1		Modelo 2		Razão	
	Tempo(s)	Objetivo	Tempo(s)	Objetivo	Tempo(s)	Objetivo
15	*	*	3600	2673	0	0
20	*	*	6.23	3223	0	0
25	*	*	1.78	3223	0	0
30	*	*	1.73	3223	0	0
35	*	*	1.87	3223	0	0
40	*	*	1.82	3223	0	0

* Estouro de memória

Fonte: Elaborada pelo autor

Figura 13 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 4 da segunda etapa de testes



Fonte: Elaborada pelo autor

5.2.5 Caso 5

Teste realizado considerando que existem doze caixas de cada tipo.

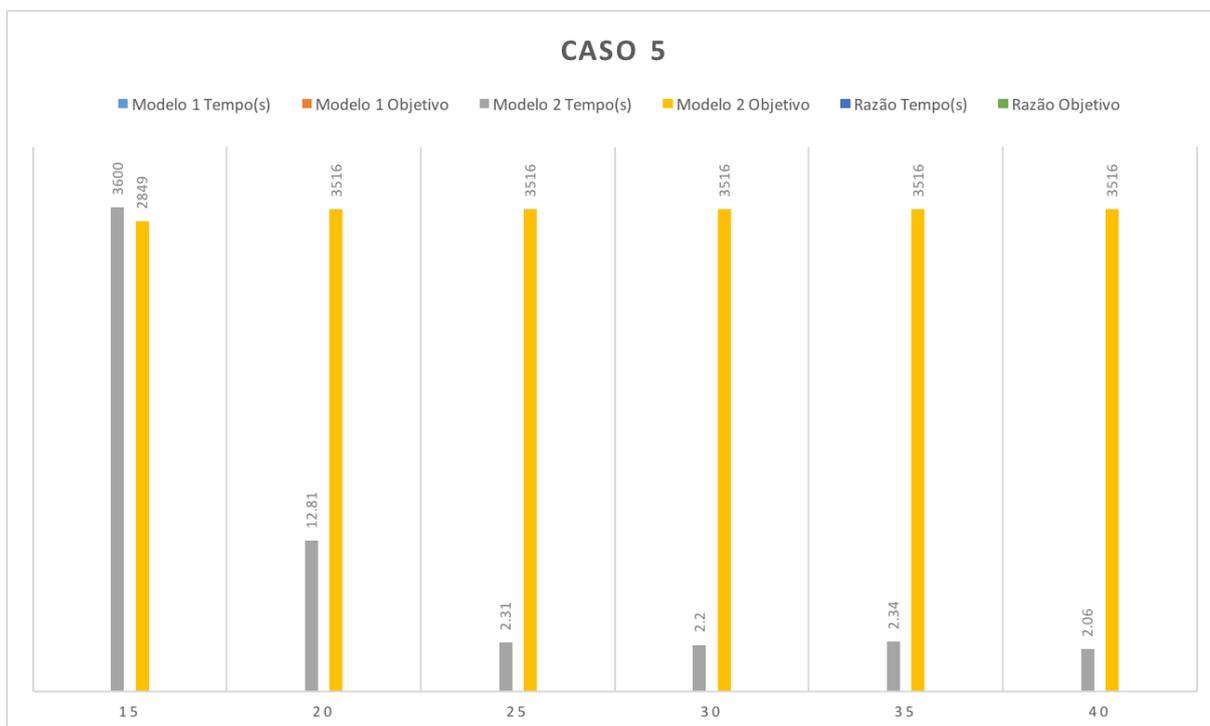
Tabela 16 – Dados coletados no Caso 5 da segunda etapa de testes

Lado do Contêiner	Modelo 1		Modelo 2		Razão	
	Tempo(s)	Objetivo	Tempo(s)	Objetivo	Tempo(s)	Objetivo
15	*	*	3600	2849	0	0
20	*	*	12.81	3516	0	0
25	*	*	2.31	3516	0	0
30	*	*	2.20	3516	0	0
35	*	*	2.34	3516	0	0
40	*	*	2.06	3516	0	0

* Estouro de memória

Fonte: Elaborada pelo autor

Figura 14 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 5 da segunda etapa de testes



Fonte: Elaborada pelo autor

5.2.6 Caso 6

Teste realizado considerando que existem treze caixas de cada tipo.

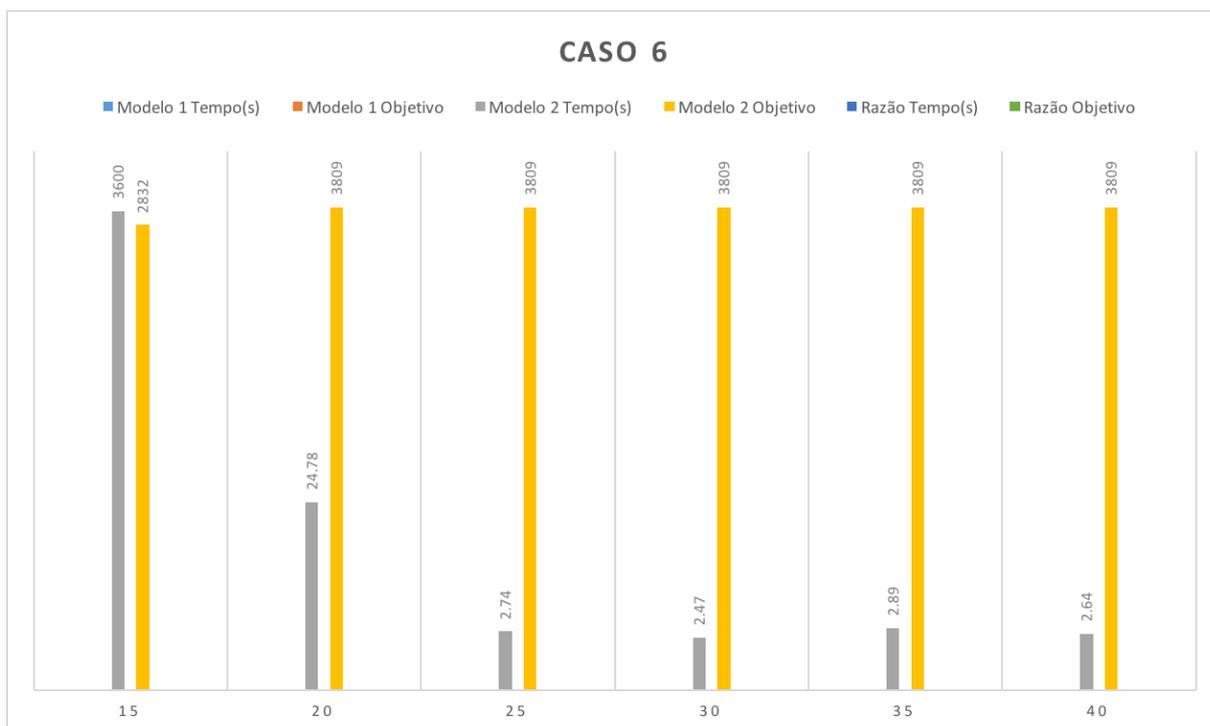
Tabela 17 – Dados coletados no Caso 6 da segunda etapa de testes

Lado do Contêiner	Modelo 1		Modelo 2		Razão	
	Tempo(s)	Objetivo	Tempo(s)	Objetivo	Tempo(s)	Objetivo
15	*	*	3600	2832	0	0
20	*	*	24.78	3809	0	0
25	*	*	2.74	3809	0	0
30	*	*	2.47	3809	0	0
35	*	*	2.89	3809	0	0
40	*	*	2.64	3809	0	0

* Estouro de memória

Fonte: Elaborada pelo autor

Figura 15 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 6 da segunda etapa de testes



Fonte: Elaborada pelo autor

Através da análise das tabelas e dos gráficos anteriores pode-se observar que o Modelo 2 foi, em todos os testes, mais eficiente do que o Modelo 1, tanto no quesito de tempo de execução quanto no valor final da função objetivo; e que o Modelo 1 excedeu a memória do computador que estava sendo utilizado para os testes em todas as etapas de cada um dos casos. Devido a isso os valores de tempo de execução e os valores finais da função objetivo relacionados ao Modelo 1 foram anulados.

5.3 O terceiro conjunto de testes

Para a terceira etapa de testes, as dimensões do contêiner foram variadas de 5 a 10. O número de tipos de caixas por caso é representado na Tabela 18.

Tabela 18 – Quantidade de tipos de caixa relacionada a cada caso da terceira etapa de testes

Caso	Quantidade de tipos de caixas
1	3
2	4
3	5
4	6
5	7
6	8

Fonte: Elaborada pelo autor

Observou-se o tempo de execução, o valor final da função objetivo e o volume total de caixas que foi excluído de cada modelo executado e ao final foi calculada a razão entre os valores obtidos pelo o modelo 1 e os valores obtidos pelo modelo 2. Os dados coletados são encontrados nas tabelas seguintes.

5.3.1 Caso 1

Teste realizado considerando que existem duas caixas de cada tipo.

Tabela 19 – Dados coletados no Caso 1 da terceira etapa de testes

Lado do Contêiner	Modelo 1			Modelo 2			Razão		
	Tempo (s)	Objetivo	Volume Excluído	Tempo (s)	Objetivo	Volume excluído	Tempo (s)	Objetivo	Volume Excluído
5	3600	85	794	0.8	93	786	4500	0.91	1.01
6	3600	153	726	3600	216	663	1	0.71	1.10
7	3600	274	605	3600	315	564	1	0.87	1.07
8	3600	347	532	3600	482	397	1	0.72	1.34
9	3600	400	479	3600	635	244	1	0.63	1.96
10	3600	588	291	3600	815	64	1	0.72	4.55

Fonte: Elaborada pelo autor

Figura 16 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 1 da terceira etapa de testes



Fonte: Elaborada pelo autor

5.3.2 Caso 2

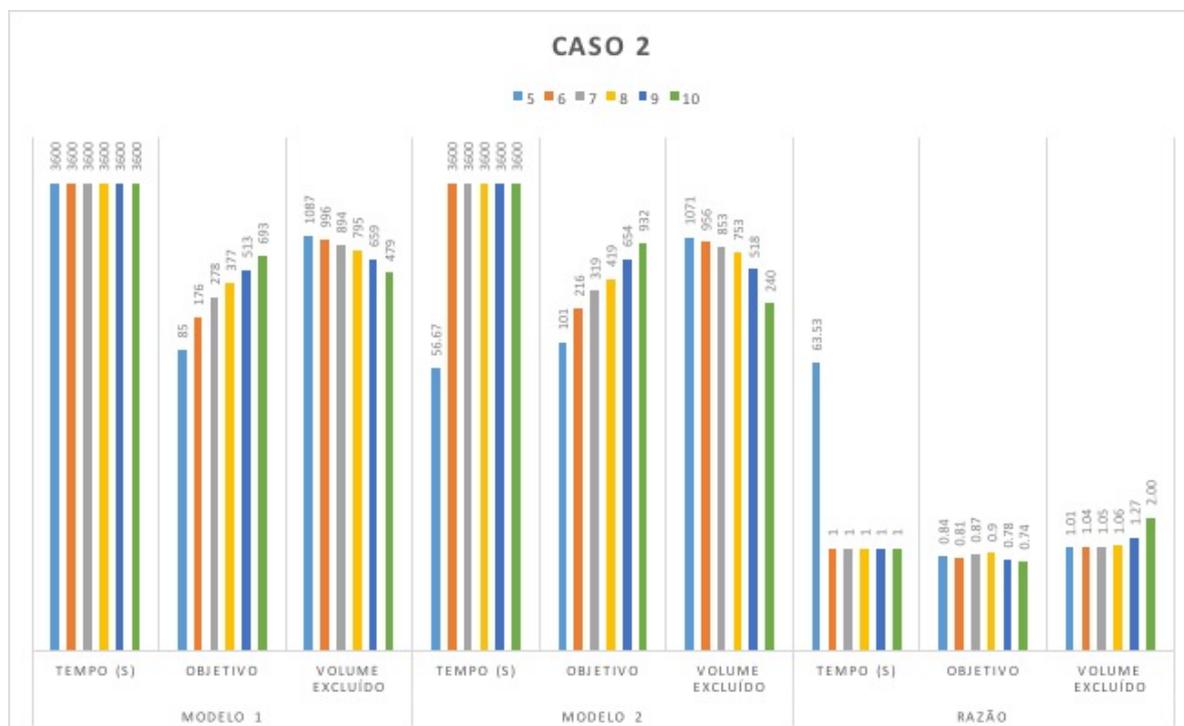
Teste realizado considerando que existem três caixas de cada tipo.

Tabela 20 – Dados coletados no Caso 2 da terceira etapa de testes

Lado do Contêiner	Modelo 1			Modelo 2			Razão	
	Tempo (s)	Objetivo	Volume Excluído	Tempo (s)	Objetivo	Volume excluído	Objetivo	Volume Excluído
5	3600	85	1087	56.67	101	1071	0.84	1.01
6	3600	176	996	3600	216	956	1.00	1.04
7	3600	278	894	3600	319	853	1.00	1.05
8	3600	377	795	3600	419	753	1.00	1.06
9	3600	513	659	3600	654	518	1.00	1.27
10	3600	693	479	3600	932	240	1.00	2.00

Fonte: Elaborada pelo autor

Figura 17 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 2 da terceira etapa de testes



Fonte: Elaborada pelo autor

5.3.3 Caso 3

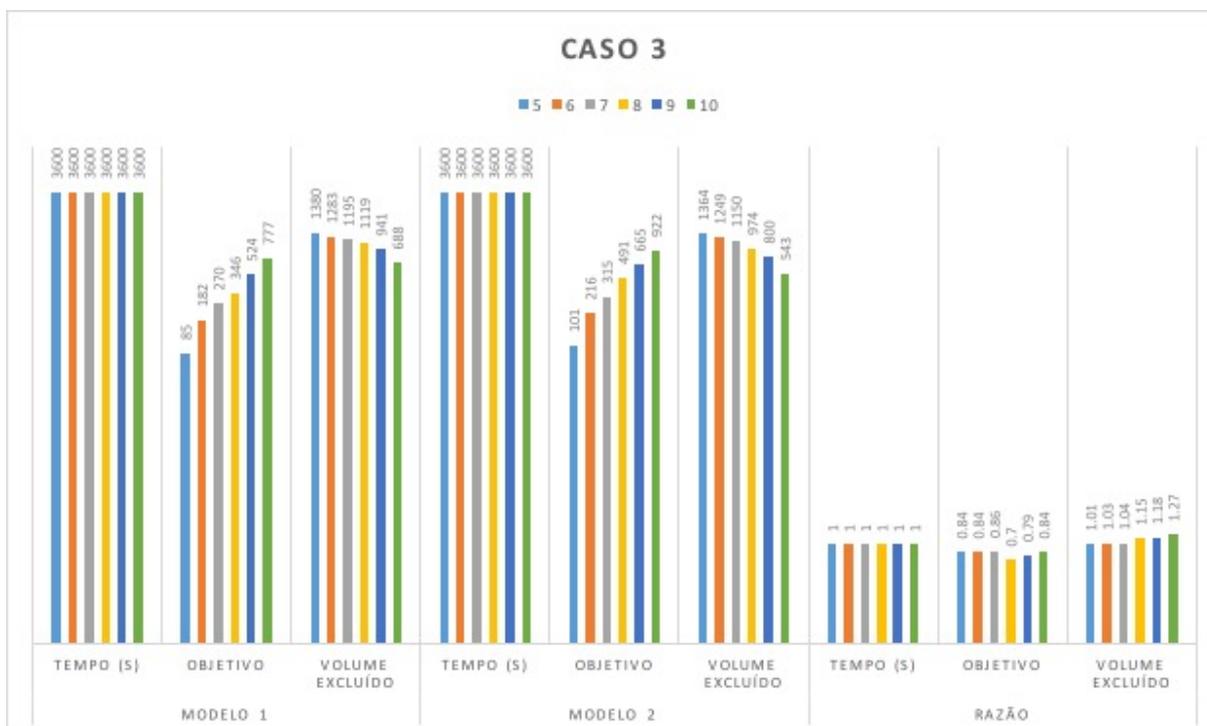
Teste realizado considerando que existem quatro caixas de cada tipo.

Tabela 21 – Dados coletados no Caso 3 da terceira etapa de testes

Lado do Contêiner	Modelo 1			Modelo 2			Razão		
	Tempo (s)	Objetivo	Volume Excluído	Tempo (s)	Objetivo	Volume excluído	Tempo (s)	Objetivo	Volume Excluído
5	3600	85	1380	3600	101	1364	1	0.84	1.01
6	3600	182	1283	3600	216	1249	1	0.84	1.03
7	3600	270	1195	3600	315	1150	1	0.86	1.04
8	3600	346	1119	3600	491	974	1	0.70	1.15
9	3600	524	941	3600	665	800	1	0.79	1.18
10	3600	777	688	3600	922	543	1	0.84	1.27

Fonte: Elaborada pelo autor

Figura 18 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 3 da terceira etapa de testes



Fonte: Elaborada pelo autor

5.3.4 Caso 4

Teste realizado considerando que existem cinco caixas de cada tipo.

Tabela 22 – Dados coletados no Caso 4 da terceira etapa de testes

Lado do Contêiner	Modelo 1			Modelo 2			Razão		
	Tempo (s)	Objetivo	Volume Excluído	Tempo (s)	Objetivo	Volume excluído	Tempo (s)	Objetivo	Volume Excluído
5	3600	85	1673	3600	101	1657	1	0.84	1.01
6	3600	206	1552	3600	216	1542	1	0.95	1.01
7	3600	256	1502	3600	313	1445	1	0.82	1.04
8	3600	422	1336	3600	496	1262	1	0.85	1.06
9	3600	479	1279	3600	632	1126	1	0.76	1.14
10	*	*	*	3600	915	843	0	0.00	0.00

* Excedeu a memória

Fonte: Elaborada pelo autor

Figura 19 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 4 da terceira etapa de testes



Fonte: Elaborada pelo autor

5.3.5 Caso 23

Teste realizado considerando que existem seis caixas de cada tipo.

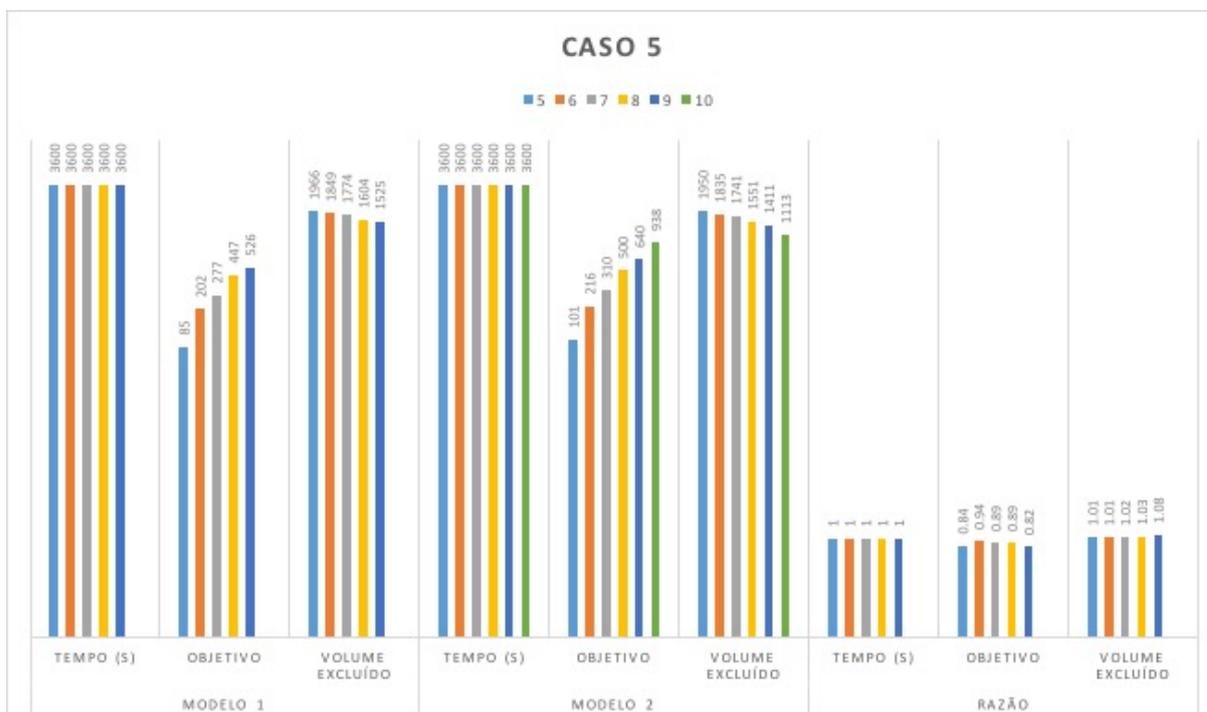
Tabela 9 – Dados coletados no Caso 5 da terceira etapa de testes

z	Modelo 1			Modelo 2			Razão		
	Tempo (s)	Objetivo	Volume Excluído	Tempo (s)	Objetivo	Volume excluído	Tempo (s)	Objetivo	Volume Excluído
5	3600	85	1966	3600	101	1950	1	0.84	1.01
6	3600	202	1849	3600	216	1835	1	0.94	1.01
7	3600	277	1774	3600	310	1741	1	0.89	1.02
8	3600	447	1604	3600	500	1551	1	0.89	1.03
9	3600	526	1525	3600	640	1411	1	0.82	1.08
10	*	*	*	3600	938	1113	0	0.00	0.00

* Excedeu a memória

Fonte: Elaborada pelo autor

Figura 20 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 5 da terceira etapa de testes



Fonte: Elaborada pelo autor

5.3.6 Caso 24

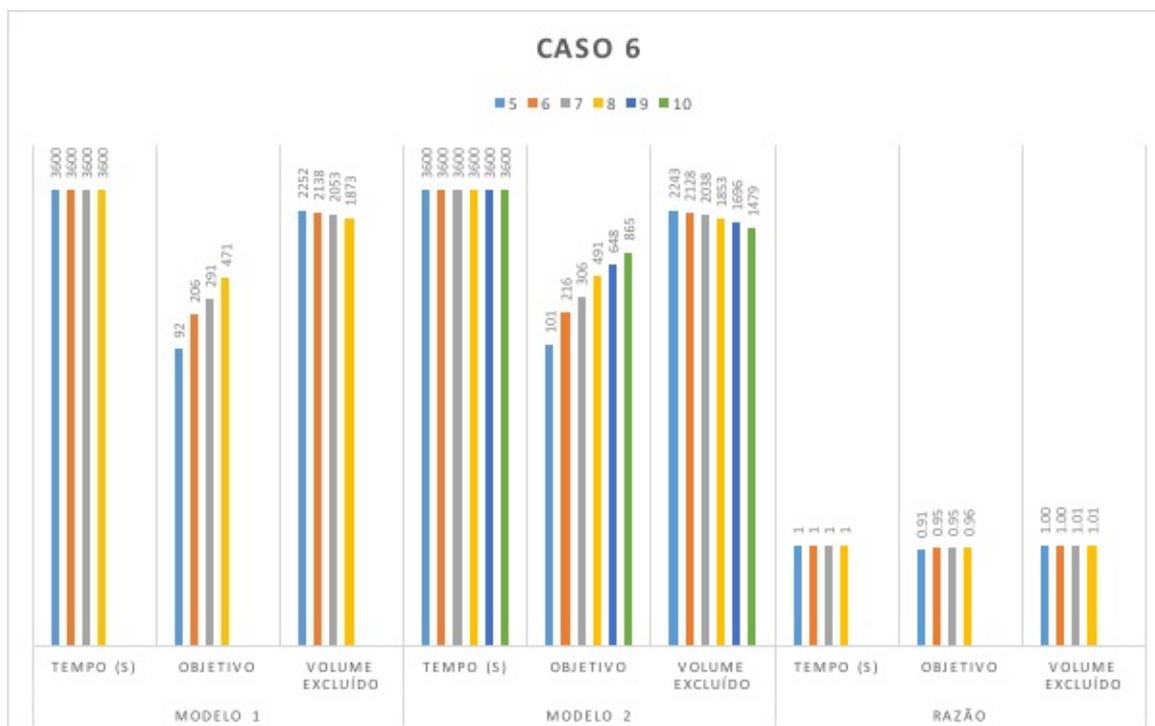
Teste realizado considerando que existem sete caixas de cada tipo.

Tabela 10 – Dados coletados no Caso 6 da terceira etapa de testes

Lado do Contêiner	Modelo 1			Modelo 2			Razão		
	Tempo (s)	Objetivo	Volume Excluído	Tempo (s)	Objetivo	Volume excluído	Tempo (s)	Objetivo	Volume Excluído
5	3600	92	2252	3600	101	2243	1	0.91	1.00
6	3600	206	2138	3600	216	2128	1	0.95	1.00
7	3600	291	2053	3600	306	2038	1	0.95	1.01
8	*	*	*	3600	491	1853	0	0.00	0.00
9	*	*	*	3600	648	1696	0	0.00	0.00
10	*	*	*	3600	865	1479	0	0.00	0.00

* Excedeu a memória
Fonte: Elaborada pelo autor

Figura 21 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 6 da terceira etapa de testes



Fonte: Elaborada pelo autor

Através da análise das tabelas e dos gráficos anteriores pode-se observar que o Modelo 2 foi, em todos os testes, mais eficiente do que o Modelo 1 em todos os quesitos (tempo de execução, valor final da função objetivo e volume de caixas excluídas) mesmo que em alguns casos ambos os modelos tenham usado o tempo máximo de 3600 segundos como critério de parada.

Vale a pena ressaltar que o Modelo 1 excedeu a memória do computador que estava sendo utilizado para os testes na parte final de cada um dos casos e que devido a isso os valores de tempo de execução, os valores finais da função objetivo e os valores referentes ao volume de caixas excluídas foram anulados.

5.4 O quarto conjunto de testes

A quarta etapa de testes também foi dividida em seis casos diferentes. As dimensões das caixas foram mantidas e os lados do contêiner variaram de 10 a 15. A Tabela 25 mostra o número de tipos de caixas por caso nesta série de testes.

Tabela 25 – Quantidade de tipos de caixa relacionada a cada caso da quarta etapa de testes

Caso	Quantidade de tipos de caixas
1	10
2	11
3	12
4	13
5	14
6	15

Fonte: Elaborada pelo autor

Observou-se o tempo de execução e o valor final da função objetivo de cada modelo executado e ao final foi calculada a razão entre os valores obtidos pelo Modelo 1 e os valores obtidos pelo Modelo 2. Os dados coletados são encontrados nas tabelas seguintes.

5.4.1 Caso 1

Teste realizado considerando que existem oito caixas de cada tipo.

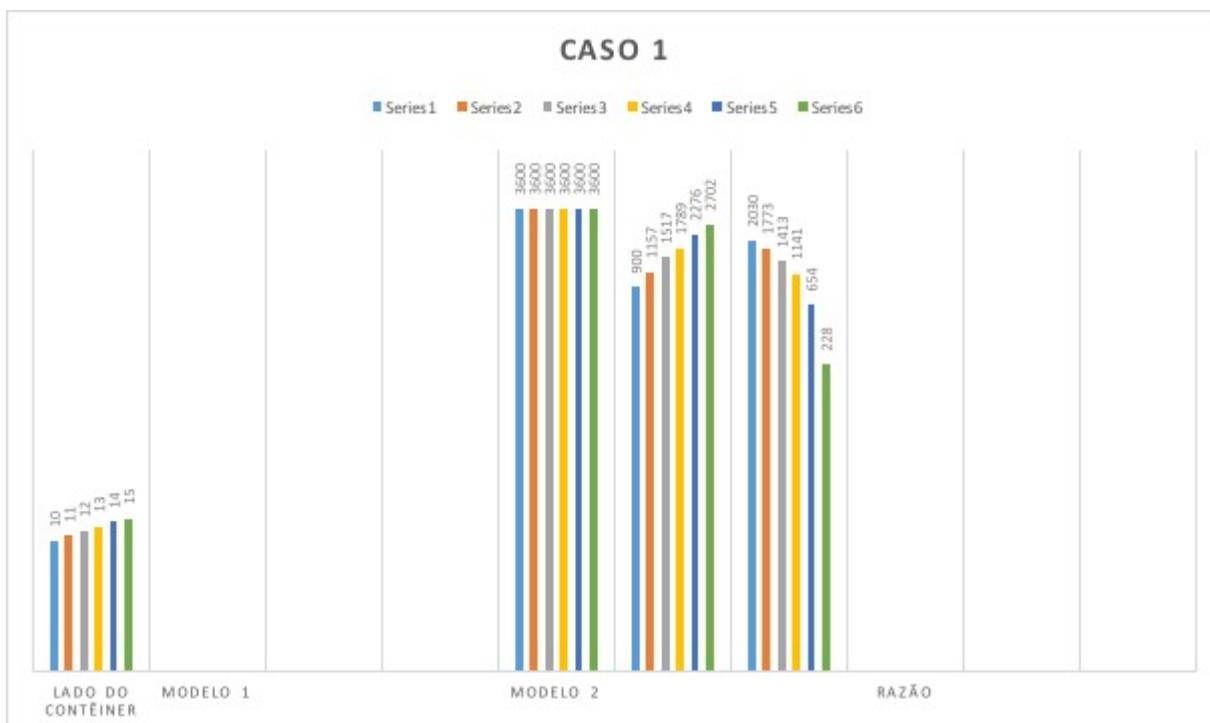
Tabela 26 – Dados coletados no Caso 1 da quarta etapa de testes

Lado do Contêiner	Modelo 1			Modelo 2			Razão		
	Tempo (s)	Objetivo	Volume Excluído	Tempo (s)	Objetivo	Volume excluído	Tempo (s)	Objetivo	Volume Excluído
10	*	*	*	3600	900	2030	0.00	0.00	0.00
11	*	*	*	3600	1157	1773	0.00	0.00	0.00
12	*	*	*	3600	1517	1413	0.00	0.00	0.00
13	*	*	*	3600	1789	1141	0.00	0.00	0.00
14	*	*	*	3600	2276	654	0.00	0.00	0.00
15	*	*	*	3600	2702	228	0.00	0.00	0.00

* Excedeu a memória

Fonte: Elaborada pelo autor

Figura 22 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 1 da quarta etapa de testes



Fonte: Elaborada pelo autor

5.4.2 Caso 2

Teste realizado considerando que existem nove caixas de cada tipo.

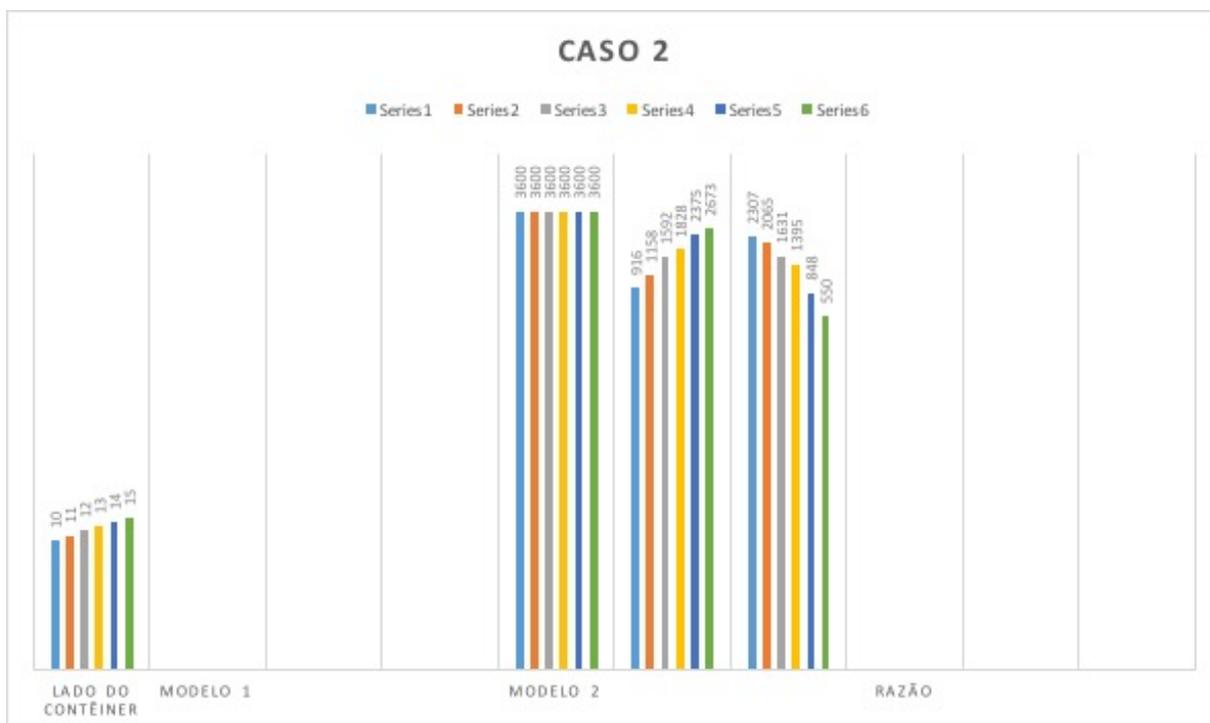
Tabela 27 – Dados coletados no Caso 2 da quarta etapa de testes

Lado do Contêiner	Modelo 1			Modelo 2			Razão		
	Tempo (s)	Objetivo	Volume Excluído	Tempo (s)	Objetivo	Volume excluído	Tempo (s)	Objetivo	Volume Excluído
10	*	*	*	3600	916	2307	0.00	0.00	0.00
11	*	*	*	3600	1158	2065	0.00	0.00	0.00
12	*	*	*	3600	1592	1631	0.00	0.00	0.00
13	*	*	*	3600	1828	1395	0.00	0.00	0.00
14	*	*	*	3600	2375	848	0.00	0.00	0.00
15	*	*	*	3600	2673	550	0.00	0.00	0.00

* Excedeu a memória

Fonte: Elaborada pelo autor

Figura 23 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 2 da quarta etapa de testes



Fonte: Elaborada pelo autor

5.4.3 Caso 3

Teste realizado considerando que existem dez caixas de cada tipo.

Tabela 28 – Dados coletados no Caso 3 da quarta etapa de testes

Lado do Contêiner	Modelo 1			Modelo 2			Razão		
	Tempo (s)	Objetivo	Volume Excluído	Tempo (s)	Objetivo	Volume excluído	Tempo (s)	Objetivo	Volume Excluído
10	*	*	*	3600	857	2659	0.00	0.00	0.00
11	*	*	*	3600	1182	2334	0.00	0.00	0.00
12	*	*	*	3600	1528	1988	0.00	0.00	0.00
13	*	*	*	3600	1834	1682	0.00	0.00	0.00
14	*	*	*	3600	2370	1146	0.00	0.00	0.00
15	*	*	*	3600	2834	682	0.00	0.00	0.00

* Excedeu a memória

Fonte: Elaborada pelo autor

Figura 24 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 3 da quarta etapa de testes



Fonte: Elaborada pelo autor

5.4.4 Caso 4

Teste realizado considerando que existem onze caixas de cada tipo.

Tabela 29 – Dados coletados no Caso 4 da quarta etapa de testes

Lado do Contêiner	Modelo 1			Modelo 2			Razão		
	Tempo (s)	Objetivo	Volume Excluído	Tempo (s)	Objetivo	Volume excluído	Tempo (s)	Objetivo	Volume Excluído
10	*	*	*	3600	876	2933	0.00	0.00	0.00
11	*	*	*	3600	1184	2625	0.00	0.00	0.00
12	*	*	*	3600	1536	2273	0.00	0.00	0.00
13	*	*	*	3600	1897	1912	0.00	0.00	0.00
14	*	*	*	3600	2357	1452	0.00	0.00	0.00
15	*	*	*	3600	2832	977	0.00	0.00	0.00

* Excedeu a memória

Fonte: Elaborada pelo autor

Figura 25 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 4 da quarta etapa de testes



Fonte: Elaborada pelo autor

5.4.5 Caso 5

Teste realizado considerando que existem doze caixas de cada tipo.

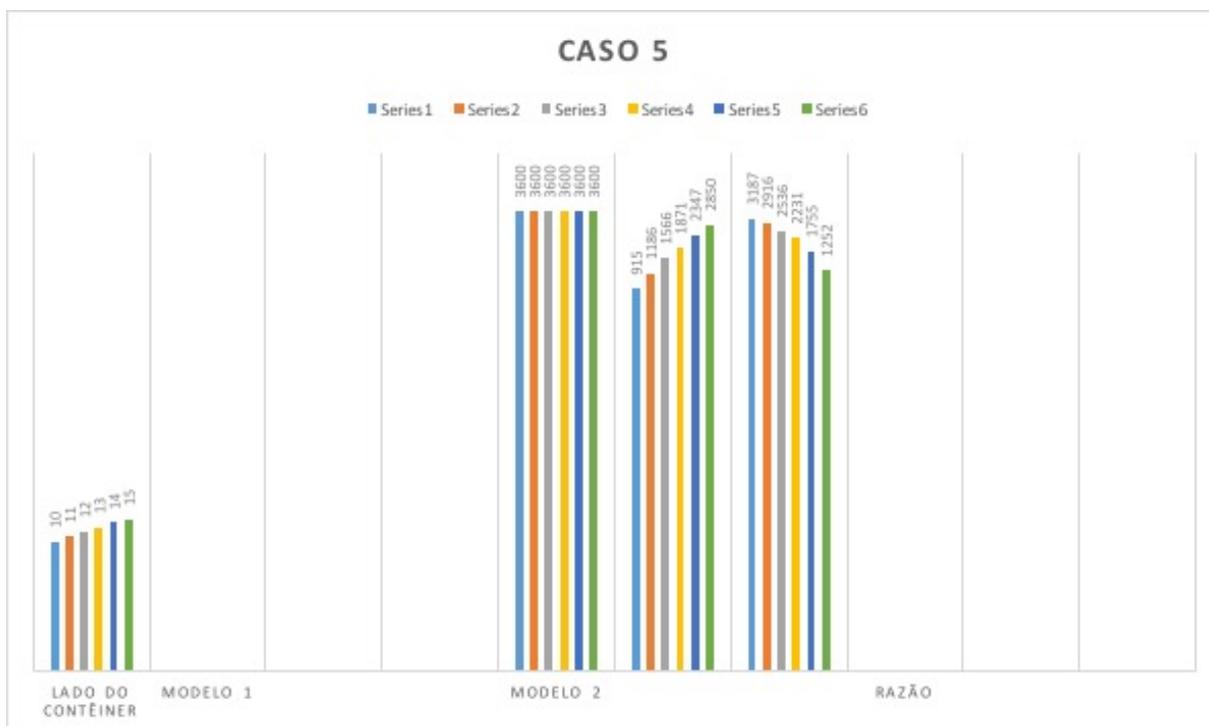
Tabela 30 – Dados coletados no Caso 5 da quarta etapa de testes

Lado do Contêiner	Modelo 1			Modelo 2			Razão		
	Tempo (s)	Objetivo	Volume Excluído	Tempo (s)	Objetivo	Volume excluído	Tempo (s)	Objetivo	Volume Excluído
10	*	*	*	3600	915	3187	0.00	0.00	0.00
11	*	*	*	3600	1186	2916	0.00	0.00	0.00
12	*	*	*	3600	1566	2536	0.00	0.00	0.00
13	*	*	*	3600	1871	2231	0.00	0.00	0.00
14	*	*	*	3600	2347	1755	0.00	0.00	0.00
15	*	*	*	3600	2850	1252	0.00	0.00	0.00

* Excedeu a memória

Fonte: Elaborada pelo autor

Figura 26 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 5 da quarta etapa de testes



Fonte: Elaborada pelo autor

5.4.6 Caso 6

Teste realizado considerando que existem treze caixas de cada tipo.

Tabela 31 – Dados coletados no Caso 6 da quarta etapa de testes

Lado do Contêiner	Modelo 1			Modelo 2			Razão		
	Tempo (s)	Objetivo	Volume Excluído	Tempo (s)	Objetivo	Volume excluído	Tempo (s)	Objetivo	Volume Excluído
10	*	*	*	3600	930	3465	0.00	0.00	0.00
11	*	*	*	3600	1210	3185	0.00	0.00	0.00
12	*	*	*	3600	1505	2890	0.00	0.00	0.00
13	*	*	*	3600	1878	2517	0.00	0.00	0.00
14	*	*	*	3600	2402	2093	0.00	0.00	0.00
15	*	*	*	3600	2775	1620	0.00	0.00	0.00

* Excedeu a memória

Fonte: Elaborada pelo autor

Figura 27 – Relação entre o tempo de execução e o valor final da função objetivo no Caso 6 da quarta etapa de testes



Fonte: Elaborada pelo autor

Através da análise das tabelas e dos gráficos anteriores pode-se observar que o Modelo 2 foi, em todos os testes, mais eficiente do que o Modelo 1 pelo fato de que o Modelo 1 excedeu a memória do computador que estava sendo utilizado para os testes em todas as etapas de cada um dos casos. Devido a isso os valores de tempo de execução, os valores finais da função objetivo e os valores referentes ao volume de caixas excluídas relacionados ao Modelo 1 foram anulados.

6 CONCLUSÃO E TRABALHOS FUTUROS

6.1 Conclusão

Esse trabalho apresenta um modelo matemático mais eficiente que o apresentado previamente para a resolução do problema de carregamento de contêineres em três dimensões com considerações de não-sobreposição das caixas.

O modelo matemático proposto foi definido utilizando técnicas de Programação Linear Inteira Mista que apresenta variáveis binárias e reais.

Como pôde ser observado no capítulo anterior, que apresenta tabelas e gráficos contendo os dados relacionados aos testes de ambos os modelos, quando expostos aos mesmos dados de entrada, pode-se concluir que o objetivo do trabalho foi atingido, ou seja, o modelo desenvolvido é, de fato, mais eficiente que o apresentado na literatura.

Tal eficiência se dá ao fato de que o número de variáveis e restrições utilizadas pelo modelo desenvolvido (Modelo 2) são muito menores que o modelo presente na literatura (1). Apesar de ambos os modelos utilizarem a técnica de comparação de caixas por grupos de dois para garantir a não sobreposição das caixas, O Modelo 2 acaba por ser mais eficiente, devido ao fato de que este utiliza apenas quatro variáveis diferentes para cada caixa, enquanto o Modelo 1 utiliza $(L - l_i) \cdot (W - w_i) \cdot (H - h_i)$ variáveis para cada caixa do tipo i . Levando isso em consideração, podemos concluir que a complexidade do Modelo 1 aumenta exponencialmente para casos no qual o contêiner utilizado possui dimensões muito grandes e as caixas possuem dimensões muito pequenas (que é considerada uma situação normal quando se trata o problema de carregamento de contêineres).

Esse problema não é encontrado no Modelo 2, pois apenas quatro variáveis são relacionadas a cada caixa e o grau de complexidade deste está apenas relacionado ao o número de caixas que se precisa organizar, ou seja, quanto maior a quantidade de caixas, maior será a sua complexidade. Porém, como apresentado na segunda etapa de testes do capítulo anterior, o Modelo 2 funciona perfeitamente para casos complexos (como o citado acima) enquanto o Modelo 1 excedeu a memória do computador utilizado diversas vezes, tanto para casos de complexidade média quanto para os casos com complexidade alta.

6.2 Trabalhos futuros

Como possíveis trabalhos futuros pode-se apontar a implementação das seguintes restrições para serem acrescentadas ao modelo desenvolvido:

- Múltiplos destinos, que leva em consideração que caixas que devem ser entregues para destinos iguais devem ser posicionadas perto umas das outras para facilitar a retirada das mesmas;
- Empilhamento, que leva em consideração a pressão máxima que cada caixa pode suportar ao haver outras caixas empilhadas em cima delas;
- Estabilidade, que leva em consideração que as caixas, ao serem alocadas, precisam ter suas faces (laterais, superiores e/ou inferiores) suportadas por outras caixas.

Seria interessante que comparações fossem feitas entre os diferentes modelos conforme restrições fossem sendo acrescentadas, para analisar o aumento na complexidade do modelo e qual a interferência nos resultados coletados nos testes.

7 REFERÊNCIAS

ANDRADE, E. L. **Introdução à pesquisa operacional**. 2 ed. Rio de Janeiro: LCT, 1998.

ARAUJO, O.C.B.; ARMENTANO, V.A. (2007). **A multi-start random constructive heuristic for the container loading problem**. *Pesquisa Operacional*, v.27, n.2, p.311-331. Disponível em: < http://www.scielo.br/scielo.php?pid=S0101-74382007000200007&script=sci_arttext>. Acesso em: 10 jun. 2015.

BARBOZA, A. L. **Simulação e técnicas da computação evolucionária aplicadas a problemas de Programação Linear Inteira Mista**. 2005. Tese (Doutorado) - Universidade Tecnológica Federal do Paraná, PR, Brasil, 2005. Disponível em < <http://repositorio.utfpr.edu.br/jspui/handle/1/74>>. Acesso em: 1 dez. 2015.

BARRA, C. de A. (2008). **Otimização do processo de distribuição de locomotivas**.

BEASLEY, J.E. (1985). **An exact two-dimensional non-guillotine cutting tree search procedure**. *Operations Research*, v.33, n.1, p.49-64. Disponível em: < https://www.researchgate.net/publication/242504207_An_Exact_Two-Dimensional_Non-Guillotine_Cutting_Tree_Search_Procedure>. Acesso em: 15 set. 2015.

BIANCHI, R.C.; SALDANHA, M.D. **Gerenciamento do estoque na cadeia de suprimentos de uma empresa de serviços e varejo de pneus**. Centro Universitário Franciscano (UNIFRA) Santa Maria - RS, 2004. Disponível em: < http://www.ead.fea.usp.br/Semead/7semead/paginas/artigos%20recebidos/Opera%E7oes/OP09_-_Gerenciamento_do_estoque.PDF> Acesso em: 8 set. 2015.

BISCHOFF, E.E.; RATCLIFF, M.S.W. (1995). **Issues in the development of approaches to container loading**. *Omega*, v.23, n.4, p.377-390. Disponível em: < <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.468.7981&rep=rep1&type=pdf>>. Acesso em: 10 set. 2015.

BORTFELDT, A. & GEHRING, H. (2001). **A hybrid genetic algorithm for the container loading problem**. *European Journal of Operational Research*, v.131, n. 1, p.143-161. Disponível em: < <http://www.sciencedirect.com/science/article/pii/S0377221700000552>>. Acesso em: 15 nov. 2015.

BORTFELDT, A. & WÄSCHER, G. (2013). **Constraints in container loading – A state-of-art review**. *European Journal of Operational Research*, v.229, n.1, p.1-20. Disponível em: < <http://www.sciencedirect.com/science/article/pii/S037722171200937X>>. Acesso em: 10 out. 2015.

BOWERSOX, D.J.; CLOSS, D.J. (2001). **Logística empresarial: o processo de integração da cadeia de suprimento**. Atlas, São Paulo. 594p. Disponível em: < https://books.google.com.br/books/about/Log%C3%ADstica_empresarial.html?hl=pt-BR&id=XSt1AAAACAAJ>. Acesso em: 18 out. 2015.

CHEN, C.S.; LEE, S.M.; SHEN, Q.S. (1995). **An analytical model for the container loading problem**. *European Journal of Operational Research*, v.80, n.1, p.68-76. Disponível em: < <http://libra.msra.cn/Publication/2085017/an-analytical-model-for-the-container-loading-problem>>. Acesso em: 27 out. 2015.

ELEY, M. (2003). **A bottleneck assignment approach to the multiple container loading problem**. *OR Spectrum*, v.25, n.1, p.45-60. Disponível em: < <http://link.springer.com/article/10.1007%2Fs002910200113>>. Acesso em: 13 ago. 2015.

GARCIA, S.; GUERREIRO, R.; CORRAR, L. J. (1997). **Teoria das Restrições e Programação Linear**. V Congresso Internacional de Custos. Acapulco, México.

GILMORE, P.C.; GOMORY, R.E. (1961). **A linear programming approach to the cutting stock problem**. *Operations Research*. v.9, n.6, p.849-859. Disponível em: < <http://www4.ncsu.edu/~kksivara/ma505/handouts/gilmore-gomory1.pdf>>. Acesso em: 25 ago. 2015.

GILMORE, P.C.; GOMORY, R.E. (1963). **A linear programming approach to the cutting stock problem - Part II**. *Operations Research*. v.11, n.6, p.863-888. Disponível em: < <http://www4.ncsu.edu/~kksivara/ma505/handouts/gilmore-gomory2.pdf>>. Acesso em: 25 ago. 2015.

GILMORE, P.C.; GOMORY, R.E. (1965). **Multistage cutting stock problems of two and more dimensions**. *Operations Research*, v.13, n.1, p.94-120. Disponível em: < <http://people.math.yorku.ca/chensy/GG1964.pdf>>. Acesso em: 13 out. 2015.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização Combinatória e Programação Linear**, 2. ed. São Paulo: Editora Campus LTDA (2000).

JUNQUEIRA, L.; MORABITO, R.; YAMASHITA, D. S. **Modelos de Otimização para problemas de carregamento de contêineres com considerações de estabilidade e empilhamento**. Departamento de Engenharia de Produção Universidade Federal de São Carlos (UFSCar) São Carlos – SP, 2009. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-74382010000100005> Acesso em: 2 abr. 2015.

LISBOA, E. **Apostila de Pesquisa Operacional**, 2002. Disponível em: <http://www.decom.ufop.br/prof/rduarte/CIC271/apostila_po.pdf> Acesso em: 10 nov. 2015.

MARTELLO, S.; PISINGER, D.; Vigo, D. (2000). **The three-dimensional bin packing problem**. Operations Research, v.48, n.2, p.256-267. Disponível em: < <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.47.2512>>. Acesso em: 18 set. 2015.

MARTINS, E. M. de O.; SILVA, F. F. da; VIEIRA, A. S.; VENTURA, A. F. A.; JÚNIOR, R. V. (2014). **Determinação do percurso mínimo para coleta seletiva de resíduos sólidos na cidade de Sousa no sertão do estado da Paraíba**. Revista Eletrônica Pesquisa Operacional para o Desenvolvimento. v.6, n.2, p.268-278. Disponível em: < <http://www.podesenvolvimento.org.br/inicio/index.php?journal=podesenvolvimento&page=article&op=view&path%5B%5D=314>>. Acesso em: 27 ago. 2015.

MORABITO, R.; ARENALES, M. (1994). **An And/Or-graph approach to the container loading problem**. International Transactions in Operational Research, v.1, n.1, p.59-73. Disponível em: < <http://www.dep.ufscar.br/docentes/morabito/itor94.pdf>>. Acesso em: 17 set. 2015.

MORABITO, R. (1994). **Modelos de otimização para o problema de corte nas indústrias de papel e papelão e de móveis**. Gestão & Produção, v.1, n.1, p.59-76. Disponível em: < http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-530X1994000100004&lng=pt&nrm=iso>. Acesso em: 17 ago. 2015.

MORABITO, R.; ARENALES, M. (1997). **Abordagens para o problema do carregamento de contêineres**. Pesquisa Operacional, v.17, n.1, p.29-56. Disponível em: <<http://www.dep.ufscar.br/docentes/morabito/po97.pdf>>. Acesso em: 21 set. 2015.

MORO, L. F. L. **Técnicas de Otimização Mista-Inteira para o planejamento e programação de produção em refinarias de petróleo**. Escola Politécnica da Universidade de São Paulo, SP, Brasil, 2000.

PEREIRA, W. I. (2009). **Pesquisa operacional: ferramenta para a competitividade**. Revista Embanews, p.54-54.

PISINGER, D. (2002). **Heuristics for the container loading problem**. European Journal of Operational Research, v.141, n.2, p.382-392. Disponível em: < https://www.researchgate.net/publication/222558664_Heuristic_for_the_container_loading_problem >. Acesso em: 23 ago. 2015.

PRADO, D. **Programação linear**. 6. ed. rev. São Paulo: DG, 2012. Disponível em: < http://www.falconi.com/flcn_book/programacao-linear/> Acesso em: 10 nov. 2015.

PUCCINI, A. de L. **Introdução à programação linear**. Rio de Janeiro: S.A, 1980.

RANCK JUNIOR, R.; YANASSE, H. H.; Morabito, R. (2014). **Uma heurística para um problema de carregamento de caixas em um contêiner compartimentado**. Oficina Nacional de Problemas de Corte, Empacotamento, Dimensionamento de Lotes e Programação da Produção, São Carlos. XVI ONPCE. Disponível em: <http://www.oficina2014.dep.ufscar.br/resumos/resumo_101.pdf>. Acesso em: 23 ago. 2015.

RATCLIFF, M.S.W. & BISCHOFF, E.E. (1998). **Allowing for weight considerations in container loading**. *OR Spektrum*, v.20, n.1, p.65-71. Disponível em: <<http://link.springer.com/article/10.1007%2FBF01545534>>. Acesso em: 28 set. 2015.

SCHEITHAUER, G.; TERNO, J.; RIEHME, J.; SOMMERWEISS, U. (1996). **A new heuristic approach for solving the multi-pallet packing problem**. Technical Report, MATH-NM- 03-1996, Technische Universität Dresden, Dresden. Disponível em: <<http://www.math.tu-dresden.de/~scheith/PREPRINTS/1996-PRE.PS>>. Acesso em: 18 ago. 2015.

SILVA, E. M.; SILVA, E. M.; GONÇALVES, V.; MUROLO, A. C. **Pesquisa Operacional: programação linear**. 3. ed. São Paulo: Atlas, 1998.

SILVA, J.L.C.; SOMA, N.Y. (2003). **Um algoritmo polinomial para o problema de empacotamento de contêineres com estabilidade estática da carga**. *Pesquisa Operacional*, v.23, n.1, p.79-98. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-74382003000100007>. Acesso em: 21 set. 2015.

TEIXEIRA, V. G. **Aplicação de programação linear na alocação de vagões gôndola para o transporte de ferro gusa na mrs logística s.a.** 2011. Dissertação (Graduação) Engenharia de Produção - Universidade Federal de Juiz de Fora (UFJF) Juiz de Fora, MG, 2011. Disponível em: <http://www.ufjf.br/ep/files/2014/07/2011_3_Vinicius.pdf>. Acesso em: 21 nov. 2015.

TEIXEIRA apud Introdução à Pesquisa Operacional, 1980. Disponível em: <http://www.ufjf.br/ep/files/2014/07/2011_3_Vinicius.pdf> Acesso em: 10 nov. 2015.

TERNO, J.; SCHEITHAUER, G.; SOMMERWEISS, U.; RIEHME, J. (2000). **An efficient approach for the multi-pallet loading problem**. *European Journal of Operational Research*, v.123, n.2, p.372-381. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221799002635>>. Acesso em: 1 out. 2015.