



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

**Modelos e algoritmos para um
problema integrado de planejamento,
sequenciamento, alocação de pátio e
alocação de berço em terminais
portuários graneleiros**

João Luiz Marques de Andrade

Orientador: Gustavo Campos Menezes

BELO HORIZONTE
ABRIL DE 2021

João Luiz Marques de Andrade

**Modelos e algoritmos para um problema
integrado de planejamento, sequenciamento,
alocação de pátio e alocação de berço em
terminais portuários graneleiros**

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Matemática e Computacional do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Mestre em Modelagem Matemática e Computacional.

Área de concentração: Modelagem Matemática e Computacional

Linha de pesquisa: Sistemas Inteligentes

Orientador: Gustavo Campos Menezes

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL
BELO HORIZONTE
ABRIL DE 2021

Andrade, João Luiz Marques de
A553m Modelos e algoritmos para um problema integrado de planejamento,
sequenciamento, alocação de pátio e alocação de berço em terminais portuários
graneleiros / João Luiz Marques de Andrade. – 2021.
71 f.

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em
Modelagem Matemática e Computacional.

Orientador: Gustavo Campos Menezes.

Dissertação (mestrado) – Centro Federal de Educação Tecnológica de
Minas Gerais.

1. Terminais marítimos – Planejamento – Teses. 2. Algoritmos – Teses.
3. Lógica matemática – Teses. 4. Tecnologia da informação - Teses.
I. Menezes, Gustavo Campos. II. Centro Federal de Educação Tecnológica de
Minas Gerais. III. Título.

CDD 519.6



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
COORDENAÇÃO DO CURSO DE MESTRADO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

**“MODELOS E ALGORITMOS PARA UM PROBLEMA INTEGRADO DE
PLANEJAMENTO, SEQUENCIAMENTO, ALOCAÇÃO DE PÁTIO E
ALOCAÇÃO DE BERÇO EM TERMINAIS PORTUÁRIOS
GRANELEIROS”**

Dissertação de Mestrado apresentada por **João Luiz Marques de Andrade**, em 23 de abril de 2021, ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, e aprovada pela banca examinadora constituída pelos professores:

Prof. Dr. Gustavo Campos Menezes (Orientador)
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Carlos Renato Storck
Centro Federal de Educação Tecnológica de Minas Gerais

Prof^a. Dr^a. Elisângela Martins de Sá
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Sérgio Ricardo de Souza
Centro Federal de Educação Tecnológica de Minas Gerais

Visto e permitida a impressão,

Prof^a. Dr^a. Elizabeth Fialho Wanner
Presidenta do Colegiado do Programa de Pós-Graduação em
Modelagem Matemática e Computacional



Emitido em 26/04/2021

ATA DE DEFESA DE DISSERTAÇÃO Nº 4/2021 - DELCOM (11.58.04)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 18/05/2021 09:12)

CARLOS RENATO STORCK

PROFESSOR ENS BASICO TECN TECNOLOGICO

DELCOM (11.58.04)

Matrícula: 2110383

(Assinado digitalmente em 18/05/2021 10:32)

ELISANGELA MARTINS DE SA

PROFESSOR ENS BASICO TECN TECNOLOGICO

DCSA (11.56.04)

Matrícula: 2228059

(Assinado digitalmente em 18/05/2021 08:58)

GUSTAVO CAMPOS MENEZES

PROFESSOR ENS BASICO TECN TECNOLOGICO

DCCN (11.58)

Matrícula: 1550421

(Assinado digitalmente em 18/05/2021 12:11)

SERGIO RICARDO DE SOUZA

PROFESSOR DO MAGISTERIO SUPERIOR

DECOM (11.56.03)

Matrícula: 300700

Para verificar a autenticidade deste documento entre em <https://sig.cefetmg.br/documentos/> informando seu número:
4, ano: **2021**, tipo: **ATA DE DEFESA DE DISSERTAÇÃO**, data de emissão: **26/04/2021** e o código de verificação:
2418e0086e

Agradecimentos

Meus sinceros agradecimentos a:

A minha família, em especial os meus pais, que sempre me apoiaram e me deram todo o suporte para eu trilhar no caminho que escolhi. A Raphaela, minha namorada, que sempre esteve ao meu lado me apoiando e motivando.

Ao Prof. Dr. Gustavo Campos Menezes, pela oportunidade, dedicação, paciência, e por todo o apoio dado ao longo deste trabalho. Sempre compreensível e não medindo esforços para ajudar no que fosse preciso é com certeza uma pessoa importante no presente trabalho.

Aos professores e membros examinadores, que me deram a honra de tê-los presentes na minha banca de defesa. Em especial, aos professores do PPGMMC, que colaboram no aprimoramento dos meus conhecimentos em Pesquisa Operacional.

Ao Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro e material durante o curso.

Resumo

A integração entre os processos operacionais e logísticos é de fundamental importância para garantir uma operação eficiente e produtiva de um terminal portuário. Este trabalho estuda um problema integrado de planejamento e sequenciamento, alocação de pátio e alocação de berço em um terminal portuário graneleiro. O problema pretende definir a quantidade e destino de cada produto de entrada ou saída do terminal, alocar cada produto nos pátios, estabelecer um conjunto de rotas viáveis que garantam que os produtos sejam estocados e transportados para os berços, e determinar a sequência, o tempo de atracação e a posição de cada navio simultaneamente, minimizando os custos de operação e o tempo de serviço dos navios. Os principais objetivos são desenvolver modelos matemáticos e projetar algoritmos eficientes para solucionar o problema integrado em estudo com instâncias de larga escala. As contribuições desta pesquisa referem-se a duas formulações matemáticas para o problema integrado e um algoritmo de solução para cada formulação. Um algoritmo combina o método de geração de coluna com uma heurística de mergulho com *backtracking*, uma heurística *relax-and-fix* e o algoritmo *branch-and-cut*. O outro algoritmo combina uma heurística de mergulho com *backtracking*, uma heurística *local branching*, duas heurísticas *relax-and-fix* e uma heurística *rolling horizon* com uma estratégia de fixação de variáveis. Os resultados dos testes computacionais mostram que ambas as abordagens de solução foram capazes de oferecer um limite superior de qualidade para suas respectivas formulações para instâncias de grande porte, com destaque para o desempenho da heurística que aplica a técnica de geração de colunas.

Palavras-chave: Planejamento Integrado. Sequenciamento. Alocação de Pátio. Alocação de Berço. Geração de Coluna. Heurísticas Matemáticas.

Abstract

The integration between the operational and logistical processes is of fundamental importance to guarantee an efficient and productive operation of a port terminal. This work studies an integrated planning, scheduling, yard allocation, and berth allocation problem in a bulk port terminal. The problem is to define the quantity and destination of each product entering or leaving the terminal, allocating each product in the yards, establishing a set of viable routes that guarantee that the products are stored and transported to the berth, and determining the sequence, the berthing time and the position of each ship simultaneously, minimizing the operating costs and the service time of the ships. The main objectives are to develop mathematical models and design efficient algorithms to solve the integrated problem under study with large-scale instances. The contributions of this research refer to two mathematical formulations for the integrated problem and a solution algorithm for each formulation. An algorithm combines the column generation method with a diving heuristic with backtracking, a relax-and-fix heuristic, and the branch-and-cut algorithm. The other algorithm combines a diving heuristic with limited backtracking, two relax-and-fix heuristics, a local branching heuristic, a rolling horizon heuristic with a variable-fixing strategy. The results of the computational tests show that both approaches were able to offer an upper bound of quality for their respective formulations with large instances, with emphasis on the performance of the heuristic that applies the column generation.

Keywords: Integrated Planning. Scheduling. Berth allocation. Stockyard allocation. Column Generation. Matheuristics.

Lista de Figuras

Figura 1 – Complexo portuário e seus subsistemas.	11
Figura 2 – Exemplos de rotas x que compartilham equipamento.	12
Figura 3 – Exemplo de alocação de berço com restrições de maré.	13
Figura 4 – Exemplo de solução com preempção dos microperíodos.	21
Figura 5 – Exemplo de solução sem preempção dos microperíodos.	21
Figura 6 – Decomposição do modelo.	29
Figura 7 – Fluxograma da heurística baseada em geração de colunas.	30
Figura 8 – Fluxograma do procedimento de geração de colunas e da fase 1 da heurística de mergulho com <i>backtracking</i>	31
Figura 9 – Exemplo de árvore de enumeração da heurística de mergulho com <i>backtracking</i>	32
Figura 10 – Fluxograma da fase 2 da heurística de mergulho com <i>backtracking</i>	33
Figura 11 – Fluxograma do procedimento de fixação das variáveis v_{nt}^b	35
Figura 12 – Fluxograma do algoritmo CMH.	37
Figura 13 – Fluxograma da fase 1 da heurística de mergulho com <i>backtracking</i>	37
Figura 14 – Fluxograma da fase 2 da heurística de mergulho com <i>backtracking</i>	38
Figura 15 – Comparação dos tempos computacionais entre os três conjuntos.	53
Figura 16 – Comparação dos tempos computacionais entre os três conjuntos.	59
Figura 17 – Comparação dos valores de f_{ub} entre os três conjuntos.	60
Figura 18 – Comparação do número de rotas utilizadas entre os três conjuntos.	60

Lista de Tabelas

Tabela 1 – Definição dos conjuntos.	14
Tabela 2 – Definição dos parâmetros.	14
Tabela 3 – Definição das variáveis de decisão.	15
Tabela 4 – Algumas trabalhos de decomposição de Dantzig-Wolfe e Geração de Colunas.	26
Tabela 5 – Variáveis e parâmetros adicionais utilizados para a geração de colunas.	27
Tabela 6 – Dados utilizados para gerar todas as instâncias.	47
Tabela 7 – Dados utilizados para gerar as instâncias de teste para CGBH e PIPSAPAB-I.	48
Tabela 8 – Resultados computacionais para um conjunto de instâncias com variação oferta, demanda e estoque inicial.	50
Tabela 9 – Resultados computacionais comparando três conjuntos de instâncias com variação na alocação de produtos nas subáreas.	51
Tabela 10 – Resultados computacionais para um conjunto de instâncias com variação no número de navios.	52
Tabela 11 – Resultados computacionais para um conjunto de instâncias com variação no número de berços	54
Tabela 12 – Resultados computacionais para um conjunto de instâncias com variação no números de rotas disponíveis.	55
Tabela 13 – Dados utilizados para gerar as instâncias de teste para o algoritmo CMH e <i>PIPSAPAB-II</i>	56
Tabela 14 – Comparação de limites inferiores entre PIPSAPAB-I e PIPSAPAB-II.	57
Tabela 15 – Resultados computacionais para um conjunto de instâncias com variação no número de navios.	58
Tabela 16 – Resultados computacionais comparando o algoritmo CMH com e sem <i>local branching</i>	61

Lista de Algoritmos

Algoritmo 1 – <i>Heurística relax-and-fix de arredondamento</i>	34
Algoritmo 2 – <i>Heurística relax-and-fix para variáveis v_{nt}^b</i>	41
Algoritmo 3 – <i>Local Branching</i>	43
Algoritmo 4 – <i>Heurística relax-fix para variáveis f_{pt}^s</i>	44
Algoritmo 5 – <i>Heurística Rolling Horizon</i>	46

Sumário

1 – Introdução	1
1.1 Motivação	1
1.2 Objetivos	2
1.2.1 Objetivos específicos	3
1.3 Organização do trabalho	3
2 – Fundamentação Teórica	4
2.1 Terminais portuários	4
2.2 Problemas de otimização de terminais de contêineres	5
2.3 Problemas de otimização de terminais graneleiros	6
3 – Problema integrado de planejamento, sequenciamento, alocação de pátio e alocação de berço (PIPSAPAB)	10
3.1 Definição do problema	10
3.2 Modelos para o PIPSAPAB	13
3.2.1 Formulação matemática PIPSAPAB-I	15
3.2.2 Formulação matemática PIPSAPAB-II	20
3.2.2.1 Restrições adicionais	23
3.2.3 Considerações finais	23
4 – Abordagem de solução para o PIPSAPAB-I	25
4.1 Decomposição de Dantzig-Wolfe e Geração de Colunas	25
4.2 Formulação baseada na geração de colunas	26
4.2.1 Problema Mestre Linear Restrito	26
4.2.2 Subproblema de precificação	28
4.2.3 Heurística Baseada na Geração de Colunas	29
4.2.3.1 Heurística de Mergulho com <i>Backtracking</i>	30
4.2.3.2 Heurística <i>Relax-and-Fix</i> de Arredondamento	33
5 – Abordagem de solução para o PIPSAPAB-II	36
5.1 Introdução	36
5.2 Heurística de Mergulho com <i>Backtracking</i>	37
5.3 Heurística <i>Relax-and-Fix</i> para as variáveis v	39
5.4 <i>Local Branching</i>	41
5.5 Heurística <i>Relax-and-Fix</i> para as variáveis f_{pt}^s	44
5.6 Heurística <i>Rolling Horizon</i>	45

6 – Resultados Computacionais	47
6.1 Gerador de Instâncias	47
6.2 Resultados para CGBH na solução do PIPSAPAB-I	48
6.3 Resultados para algoritmo CMH na solução do PIPSAPAB-II	56
7 – Conclusão e trabalhos futuros	62
7.1 Conclusão	62
7.2 Trabalhos futuros	63
Referências	65

Capítulo 1

Introdução

Este capítulo fornece uma breve introdução sobre a dissertação. A seção 1.1 apresenta a motivação desta pesquisa, enquanto os objetivos estão detalhados na seção 1.2. A seção 1.3 descreve a estrutura da tese.

1.1 Motivação

O transporte marítimo está entre os mais importantes meios de transporte e navegação utilizados em todo o mundo. Apesar da invenção de novos meios de transporte, o marítimo não perdeu sua importância; pelo contrário, está cada vez mais ganhando interesse da economia mundial. De acordo com a *United Nations Conference on Trade and Development* (UNCTAD, 2020), na última década houve um crescimento de aproximadamente 40% no volume total de carga transportada por navios no mundo. Estima-se que em 2019, o volume de tráfego marítimo alcançou 11 bilhões de toneladas. As cargas secas foram responsáveis por 7,9 bilhões de toneladas, sendo que os principais granéis sólidos (carvão, minério de ferro e grãos) atingiram 40,78% deste volume. O transporte de granéis sólidos foi um dos segmentos de carga que mais cresceram entre 1980 e 2019, com um crescimento anual de cerca de 5% no volume de toneladas comercializadas, totalizando um aumento acumulado de 428% no período.

Neste cenário, os terminais portuários passaram a ser um ponto estratégico, dada sua complexidade para o desenvolvimento da cadeia logística no comércio exterior (BICHOU, 2013). De uma forma simples, os portos marítimos têm as funções de manusear e armazenar enormes volumes de produtos. Eles formam uma conexão entre os transportes terrestres e as embarcações marítimas, os fornecedores e os usuários dos produtos (HAN, 2018).

Devido ao aumento drástico do comércio marítimo, os terminais estão atingindo o limite da sua capacidade, levando ao sobrecarregamento dos portos e congestionamento de navios. Em particular, o congestionamento de navios afeta não apenas o terminal

(mão-de-obra extra, congestionamento de pátios, remanuseio), mas, também, as linhas de navegação (atrasos de navios, conexões perdidas, custos extras) e outros demais participantes da cadeia. Uma solução para o congestionamento é aumentar a capacidade, com pátios maiores e novos equipamentos, embora esse caminho seja frequentemente impedido devido à falta de espaço físico, restrições orçamentárias e questões ambientais. Além disso, sempre que o terminal puder pagar o investimento adicional, poderá levar anos para a nova infraestrutura operacional. Portanto, as melhorias na eficiência e na produtividade dos portos são cada vez mais necessárias e sistemas operacionais e logísticos eficazes podem ajudar significativamente a fazer o melhor uso da infraestrutura e recursos portuários disponíveis.

A necessidade de um gerenciamento eficiente das atividades operacionais e logística nos modernos sistemas portuários é bem reconhecida e o gerenciamento das operações dos terminais graneleiros pode se beneficiar muito com os métodos da Pesquisa Operacional. A utilização de modelos matemáticos para representar atividades dos terminais e o uso de heurísticas e algoritmos exatos para solucionar os modelos, podem resultar em um melhor e maior controle das operações (seja qual for ela), proporcionar um uso mais inteligentes dos recursos disponíveis e permitir que os profissionais responsáveis tomem melhores decisões.

Os terminais portuários são estruturas complexas que possuem vários problemas operacionais e logísticos fortemente conectados. Por exemplo, a alocação de navios nos berços afeta diretamente os pátios de estocagem, pois são as demandas dos navios que determinam quando um produto deve ser transportado para o berço de atracação. Em outro exemplo, o sequenciamento dos equipamentos afeta tanto a alocação dos produtos nos estoques como o sequenciamento dos navios, pois a sequência dos equipamentos interfere em quando e onde um produto é empilhado e recuperado, e, conseqüentemente, quando e onde o navio deve ser carregado. Portanto, solucionar estes problemas individualmente, ignorando as relações entre eles, pode gerar soluções com baixa qualidade (UNSAI; OğUZ, 2019). Os problemas obtidos a partir da integração são conhecidamente NP-difíceis e requerem avançados e complexos métodos de solução, estabelecendo, desse modo, grandes desafios para acadêmicos da área.

Neste contexto de pesquisa, o presente trabalho pretende propor soluções integradas para problemas operacionais existentes em terminais portuários de granéis sólidos (minério de ferro, carvão e grãos) do tipo exportador.

1.2 Objetivos

O tema do presente trabalho envolve o estudo do problema integrado de Planejamento, Sequenciamento, Alocação de Pátio e Alocação de Berço. O principal objetivo é desenvolver

modelos matemáticos e algoritmos de solução, bem como avaliar a qualidade dos resultados obtidos.

1.2.1 Objetivos específicos

Dentre os objetivos específicos deste trabalho, é possível citar os seguintes:

- Reformular e estender o modelo matemático proposto por [Menezes, Mateus e Ravetti \(2017\)](#), para gerar uma representação integrada dos problemas de planejamento, sequenciamento, alocação de pátio e alocação de berço;
- Propor algoritmos capazes de solucionar de forma integrada os quatro problemas;
- Resolver instâncias de grande porte e baseadas em casos reais.

1.3 Organização do trabalho

Este trabalho está organizado da seguinte forma: o Capítulo 2 faz uma revisão da literatura para problemas encontrados na otimização das operações portuárias. O Capítulo 3 é dedicado à definição do problema e à introdução de dois modelos matemáticos. O Capítulo 4 apresenta uma decomposição para a primeira formulação matemática e uma heurística baseada no método de geração de colunas. O Capítulo 5 descreve um algoritmo, que combina heurísticas matemáticas, para solucionar a segunda formulação. O Capítulo 6 discute os experimentos computacionais realizados para validar os modelos matemáticos e testar a desempenho dos algoritmos. Finalmente, o Capítulo 7 finaliza esta dissertação apresentando conclusões e as propostas de trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Nas últimas décadas, os terminais portuários têm recebido um crescente interesse da comunidade acadêmica, conseqüentemente significativos progressos foram realizados na otimização de processos operacionais e logísticos dos terminais. Este capítulo fornece uma breve revisão da literatura sobre problemas otimização em terminais portuários.

2.1 Terminais portuários

De acordo com [Ligteringen e Velsink \(2012\)](#), o tipo de carga transportada e o sentido da operação (importação, exportação, transbordo, entre outros) são características que influenciam diretamente na infraestrutura do porto. Os terminais portuários podem ser basicamente divididos em terminal de contêineres (*Container Terminal*) e terminal de carga a granel (*Bulk Terminal*).

Os terminais de granéis e contêineres consistem em áreas à beira-mar e pátios. Eles precisam realizar atividades comuns, como embarque e desembarque de cargas de navios, transporte de cargas entre áreas de armazenamento e atracação, bem como empilhamento temporário na área de armazenamento. No entanto, eles diferem muito em termos do tipo de carga. Ao contrário dos terminais de contêineres, nos quais todas as cargas são transportadas em um contêiner padronizado, nos terminais de granéis os meios de armazenamento e manuseio da carga variam para diferentes tipos de materiais. Outra diferença importante é que os terminais de granéis atuam como terminal de importação ou exportação ([UNSAI; OğUZ, 2019](#)).

Como dito anteriormente, os terminais portuários possuem problemas e características distintas, permitindo então a criação de diferentes linhas de pesquisa. Desse modo, uma parte dos esforços dos pesquisadores está focada em estudar os problemas de otimização dos terminais de contêineres, enquanto a outra está focada em solucionar os problemas de otimização dos terminais graneleiros. As subseções seguintes abordam alguns dos principais

conceitos, problemas clássicos e a revisão da literatura para terminais de contêiner e de carga a granel. Este capítulo destaca a importância de investigar os problemas de otimização relacionados aos terminais portuários, além de fornecer uma breve revisão da literatura.

2.2 Problemas de otimização de terminais de contêineres

Segundo [Murty et al. \(2005\)](#), um terminal de contêiner apresenta as duas principais funções: receber contêineres de exportação para embarcar nos navios, e descarregar contêineres de importação de embarcações para entregar ao destinatário.

Para [Guan, Yang e Zhou \(2013\)](#), as operações realizadas pelos terminal de contêineres geralmente contém quatro tarefas principais: atribuir berços para embarcações, carregar ou descarregar contêineres entre um navio e um berço do porto, por meio de guindastes portuários, transportar contêineres de cada berço para o pátio do porto por meio de máquinas apropriadas e carregar ou descarregar contêineres no pátio do porto. Todas essas tarefas podem ser otimizadas utilizando as ferramentas de Pesquisa Operacional, e assim melhorar a eficiência do porto. Os problemas obtidos são classificados na literatura como NP-difícil. [Bierwirth e Meisel \(2010\)](#), [Carlo, Vis e Roodbergen \(2014\)](#), [Said e El-Horbarty \(2015\)](#) abordam alguns desses problemas e comentam sobre a complexidade de solução.

Na literatura podem ser encontrados *surveys* que apresentam uma visão geral dos estudos da pesquisa operacional relacionados aos problemas dos terminais de contêineres. [Vis e de Koster \(2003\)](#) fornecem uma classificação e um resumo dos problemas de decisão relacionados ao transbordo de contêineres. [Steenken, Voss e Stahlbock \(2004\)](#) descrevem os processos logísticos em terminais de contêineres e analisam os métodos de otimização relacionados. Este trabalho foi posteriormente atualizado por [Stahlbock e Voss \(2008\)](#). [Bierwirth e Meisel \(2010\)](#) e [Bierwirth e Meisel \(2015\)](#) focaram nos problema de alocação de berços e sequenciamento de guindastes de cais em terminais de contêineres e discutiram os algoritmos de sequenciamento já existentes. [Fransoo e Lee \(2013\)](#) destacam a importância dos terminais de contêineres na cadeia de suprimentos global e listam relevantes desafios de pesquisa. [Carlo, Vis e Roodbergen \(2014\)](#) fazem uma visão geral da literatura dos problemas relacionados às operações de transporte interno no terminal, como também indica novos caminhos para pesquisa acadêmica na área de terminais. [Gharehgozli, Roy e Koster \(2015\)](#) relatam 177 artigos publicados entre 2008 e 2015 que consideram problemas de otimização em terminais de contêineres. [Barbosa \(2018\)](#) faz uma revisão de literatura sobre os recentes avanços do problema de alocação de berço, expondo inúmeras formulações e métodos de solução do problema, e destacando que a variedade de características e objetivos levam a infinitas abordagens.

Uma parte das pesquisas de otimização dos terminais de contêineres está direcionada para a integração de problemas operacionais. [Vacca, Salani e Bierlaire \(2013\)](#) apresentam um modelo que integra os problemas de alocação de berços e atribuição de guindastes. Para obter soluções ótimas inteiras de maneira mais eficiente, é implementado o método de geração de colunas e o algoritmo exato de *Branch-and-Price*. [Shang, Cao e Ren \(2016\)](#) também investigam a integração dos problemas de alocação de berço e atribuição de guindastes de cais. Um modelo de otimização robusto é proposto considerando tempos de *setup* para os equipamentos, propriedades dos navios e localizações específicas para algumas embarcações. As soluções são baseadas em algoritmos genéticos e heurística de inserção. [Agra e Oliveira \(2017\)](#) apresentam duas formulações matemáticas para o problema de alocação de berço e atribuição de guindaste de cais. Em uma das formulações são propostos vários aprimoramentos, como desigualdades válidas. Para solucionar o problema em questão, é aplicado o algoritmo *Branch-and-Cut*, que utiliza o modelo aprimorado e limites superiores fornecidos pela heurística *rolling horizon*. [Wang et al. \(2018\)](#) estudam um complexo problema integrado de alocação de berço, atribuição de guindaste e alocação de pátio. Os autores propuseram um modelo linear inteiro misto para o problema integrado, e, para solucionar instâncias de proporções reais em tempos mais curtos, implementam uma heurística baseada no método de geração de coluna.

2.3 Problemas de otimização de terminais graneleiros

Os terminais de cargas a granel são constituídos basicamente pelas seguintes áreas: píeres e atracadouros (onde os navios são atracados para carregamento ou descarregamento dos produtos) e pátios de armazenamentos (onde os produtos permanecem estocados para uso futuro). O porto se conecta com o território por meio das linhas ferroviárias, que são responsáveis por escoar os produtos para os consumidores ou receber produtos de fornecedores (minas, produtores agrícolas, usinas, entre outros) para o carregamento dos navios. O transporte dos produtos entre ferrovias, pátios e navios é feito por meio de equipamentos específicos para o tipo de terminal e de produtos manuseados. Alguns deles são: correias transportadoras, empilhadeiras, recuperadores, entre outros. A capacidade e a disponibilidade dos equipamentos, assim como os limites de armazenamento dos pátios, são fatores que definem a capacidade do terminal em atender os contratos com os navios. A liberação mais cedo do que o programado (*despath*) ou o atraso na liberação (*demurrage*) de um navio do terminal acarretam custos para o operador do terminal (para o caso de *demurrage*) ou uma recompensa a ser paga pelo responsável da embarcação para o terminal (para o caso de *despath*).

Quando comparado o volume de pesquisas focadas em terminais de contêineres com as de terminais graneleiros, é notável o número restrito de trabalhos sobre terminais

graneleiros. Entretanto, alguns estudos desenvolvidos para os terminais de contêineres podem ser utilizados como auxílio para novas pesquisas de terminais de carga a granel. Nesta seção faremos uma breve revisão sobre problemas de otimização dos terminais relacionados ao problema integrado proposto.

O problema de alocação de berço busca determinar a posição e o tempo de atracação de cada navio dentro de um definido horizonte de planejamento. Diferentes abordagens para esse problema foram propostas na literatura. A existência de várias restrições espaciais e temporais possibilitou uma infinidade de formulações para o problema. As restrições temporais são relacionadas principalmente ao processo de chegada das embarcações e o início do serviço, enquanto as restrições espaciais estão relacionadas principalmente ao *layout* dos berços (ROBENEK et al., 2014). Baseado no *layout* dos berços, três variantes para o problema de alocação de berços podem ser identificadas: discreto, contínuo e híbrido. No contexto dos terminais graneleiros, Barros et al. (2011) estuda um problema de alocação de berço discreto com restrições de maré e estoque. Neste trabalho, os berços são homogêneos. Uma interessante consideração deste trabalho são as restrições de maré, onde, uma vez carregados os navios, eles somente poderão partir com a maré alta. Ernst et al. (2017) também consideram os efeitos da maré, porém em um problema de alocação de berço contínuo. Umang, Bierlaire e Vacca (2013) apresentam três formulações para o problema de alocação de berço híbrido. Este trabalho utiliza uma extensão do *layout* híbrido. Eles dividem o cais em um conjunto de seções de comprimento variável, onde cada seção pode ser ocupada por no máximo um navio ou parte de um navio. Dessa forma, o navio pode ocupar várias seções, e os tempos de manuseio dependem da posição de atracação da embarcação, como também da carga a ser carregada ou descarregada. Neste estudo, nós assumimos o *layout* discreto dos berços.

O problema de alocação de berço é frequentemente integrado com outros problemas operacionais dos terminais, como nos trabalhos de Robenek et al. (2014), Tang, Sun e Liu (2016), Krimi et al. (2019), Unsal e Oğuz (2019), Kasm, Diabat e Cheng (2019) e Shang, Cao e Ren (2016), Agra e Oliveira (2017), Wang et al. (2018), De et al. (2020) para terminais de contêineres. Robenek et al. (2014) estende o trabalho de Umang, Bierlaire e Vacca (2013) e integram, ao problema de alocação de berço híbrido, o problema de alocação de pátio em um único modelo. Várias premissas realistas e específicas de terminais graneleiros foram consideradas na formulação, como a chegada dinâmica de navios, restrições de congestionamento, considerações de calado de navio e profundidade da água, restrições de capacidade de manuseio e restrições de local de armazenamento baseado no tipo da carga. O artigo propõe um modelo de programação inteira mista, e apresenta um algoritmo exato *branch-and-price* que foi capaz de gerar boas soluções para instâncias reais. No artigo de Unsal e Oğuz (2019), o problema de alocação de berço é integrado aos problemas operacionais de sequenciamento de recuperadores e alocação de produtos nos pátio. O

problema considera, além das operações, restrições de janela de maré, múltiplas subáreas de estocagem, não cruzamento de recuperadores e tamanho de navios e berços. A solução é baseada na decomposição de Benders, no qual o subproblema é solucionado via *constraint programming*. O algoritmo exato proposto mostrou-se eficiente em solucionar instâncias de proporções reais.

O gerenciamento dos pátios é uma das principais operações do porto. O uso inteligente das áreas de armazenamento permite um melhor uso do espaço, evitando gargalos, como também auxilia na redução do tempo de carregamento de navios quando integrado com as operações relacionadas. Seguindo essa lógica, [Tang, Sun e Liu \(2016\)](#) estuda um problema integrado de alocação de pátio e sequenciamento de navios, com objetivo de alcançar uma maior utilização dos pátios, reduzir as perdas de produtos e os custos de transporte. Neste trabalho, as áreas de estoque do pátio são divididas em *slots* menores, em que cada *slot* pode alocar apenas um produto. As pilhas de produtos podem se estender a mais de um *slot*, sendo necessário manter um *slot* entre as pilhas de produtos diferentes para evitar a contaminação. Em [Robenek et al. \(2014\)](#), os autores consideram a distância entre a área de armazenamento e o berço de carregamento para alocar os produtos nos pátios. A pesquisa de [Boland, Gulczynski e Savelsbergh \(2012\)](#) abrange o problema de gerenciamento das pilhas de carvão em um terminal portuário da Austrália. As decisões incluem a seleção dos equipamentos de transporte, alocação no pátio de estocagem e sincronização de todo o processo com a programação dos navios. [Rocha de Paula et al. \(2019\)](#) desenvolveram um algoritmo genético para maximizar o rendimento de um terminal de exportação de carvão. O artigo estuda um método de otimização que programa as chegadas do carvão no terminal, determina os períodos de empilhamento e recuperação das pilhas dos pátios e programa os horários de chegada e partida das embarcações.

Os trabalhos ([MENEZES; MATEUS; RAVETTI, 2016](#)) e ([MENEZES; MATEUS; RAVETTI, 2017](#)) estudam um problema integrado de planejamento, sequenciamento e alocação de pátio. O problema considera planejar e sequenciar o fluxo de produtos entre os pontos de oferta, armazenamento e demanda de um terminal portuário graneleiro. Os destaques do trabalho são: a nova formulação modelo matemática e as abordagens de solução. No qual, em ([MENEZES; MATEUS; RAVETTI, 2016](#)) propõe uma heurística hierárquica e em ([MENEZES; MATEUS; RAVETTI, 2017](#)) um algoritmo *branch-and-price*. Em ambas abordagens foi possível obter bons limites superiores, ou até mesmo solução ótima, para instâncias baseadas em casos reais.

Em [Hu e Yao \(2012\)](#), os autores destacam a importância da produtividade dos empilhadores/recuperadores na eficiência operacional dos terminais de carga a granel. Neste sentido propõe um modelo de programação inteiro misto para o problema de programação de empilhadeiras/recuperadores para reduzir o tempo máximo de conclusão

de processos. Uma vez que o problema é de natureza NP-difícil, foi desenvolvido um algoritmo genético para se ter soluções de melhor qualidade em um menor tempo. [Unsal \(2020\)](#) estuda um complexo e individual problema de sequenciamento de recuperadores. O problema considera inúmeras características reais de um terminal graneleiro, como estrutura real de pátio com múltiplos carregadores, restrições de não cruzamento de recuperadores nos trilhos, tempos de *setup* e operações de empilhamento. [Angelelli et al. \(2016\)](#), [Wang, Lu e Sitters \(2018\)](#), [Burdett et al. \(2019\)](#) são também importantes trabalhos que estudam sobre problemas de sequenciamento de equipamentos de terminais graneleiros.

Capítulo 3

Problema integrado de planejamento, sequenciamento, alocação de pátio e alocação de berço (PIPSAPAB)

Este capítulo descreve o problema integrado de planejamento, sequenciamento, alocação pátio e alocação de berço em terminais portuários exportadores de granéis sólidos. A seguir, será definido o problema e apresentada duas formulações matemáticas.

3.1 Definição do problema

No Brasil, os terminais de graneis sólidos concentram a maior movimentação de cargas dos portos no país. Dados da Agência Nacional de Transporte Aquaviários ([ANTAQ, 2019](#)) mostram que os granéis sólidos representaram 64% da movimentação total das instalações portuárias em 2018, com 712 milhões de toneladas movimentadas.

Os terminais de granéis sólidos são responsáveis por movimentar cargas secas em grande volume, no qual as principais cargas são de origem agrícola e mineral. Destacam-se no Brasil a movimentação das seguintes *commodities* em granel agrícola: grão de soja, farelo de soja, milho, açúcar e outros cereais. Em granel mineral como: minério de ferro, carvão, cimento, alumina e bauxita ([VIEIRA, 2018](#)).

O minério de ferro é o principal produto comercializado. Segundo o relatório da [ANTAQ \(2019\)](#), o minério de ferro representa 33% do total da movimentação em 2019, com 367,8 milhões de toneladas movimentadas. Este produto é manipulado basicamente em três tipos: Granulado, *Sinter* e Pelota. Para cada tipo existe uma série de derivados, que se diferenciam de acordo com as características químicas (quantidade de ferro, alumina, sílica, manganês e umidade) e físicas (granularidade). O problema proposto neste trabalho considera importantes problemas operacionais de um terminal portuário graneleiro que

manipula minérios de ferro. O terminal em estudo é do tipo exportador. O complexo portuário estudado neste trabalho pode ser representado como um conjunto de três subsistemas: recepção, pátios e berços. A Figura 1 fornece uma visão geral do terminal portuário com os três subsistemas.

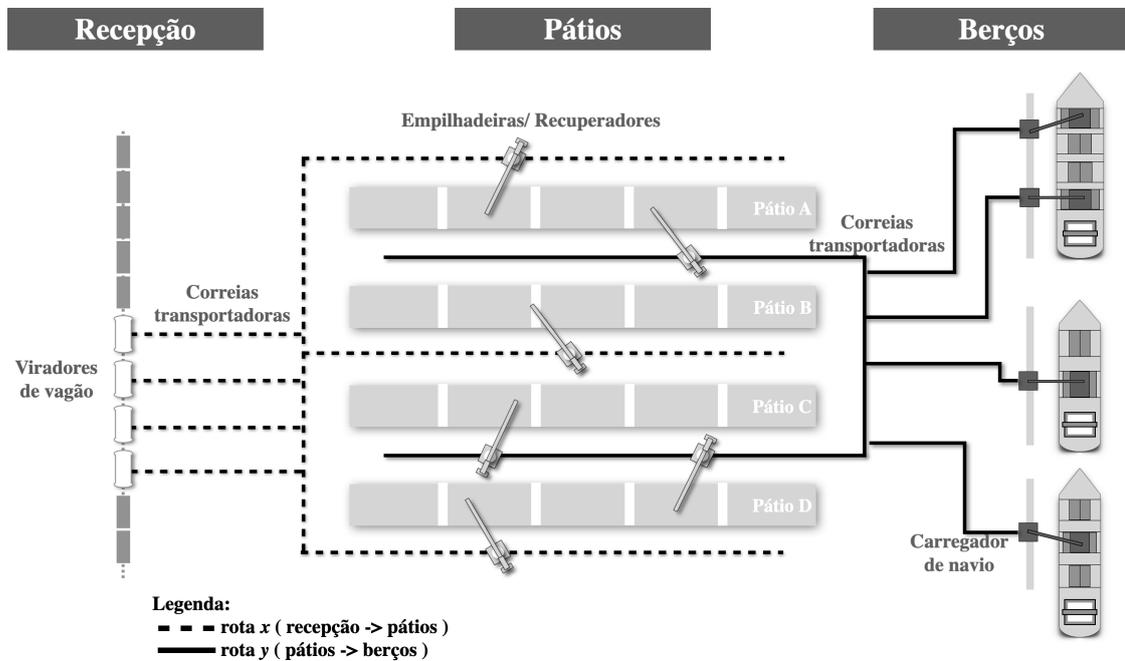


Figura 1 – Complexo portuário e seus subsistemas.

Os produtos são transportados entre os subsistemas por meio de rotas, no qual as rotas x estocam os produtos da recepção nos pátios, e as rotas y recuperam os produtos do pátio para carregar os navios. Essas rotas são encarregadas de transportar lotes de cada produto e definem várias opções de caminho por onde os produtos podem ser transportados. Desse modo, o objetivo é definir a quantidade e o destino de cada produto, e, além disso, determinar quais rotas irão transportar os produtos.

As rotas são constituídas por uma combinação de equipamentos. Diferente dos terminais de contêineres em que toda carga é padronizada, os terminais graneleiros requerem equipamentos específicos para tipos de cargas e navios. Neste estudo serão considerados os seguintes equipamentos para manusear o minério de ferro: viradores de vagão, correias transportadoras, empilhadeiras, recuperadores e carregadores de navios. Os equipamentos que estabelecem uma rota são definidos previamente. A quantidade de equipamentos é limitada e cada um deles possui uma capacidade de transporte por hora. O equipamento de menor capacidade da rota define a capacidade da mesma. Um determinado equipamento pode ser compartilhado por mais de uma rota. Caso um par de rotas compartilhem algum equipamento, estas rotas não podem ficar ativas simultaneamente. O objetivo é ativar e sequenciar o melhor conjunto de rotas que podem funcionar ao mesmo tempo para

transportar os produtos. A Figura 2 exibe um exemplo de duas rotas x (rota 1 e 2) que compartilham um empilhador/recuperador.

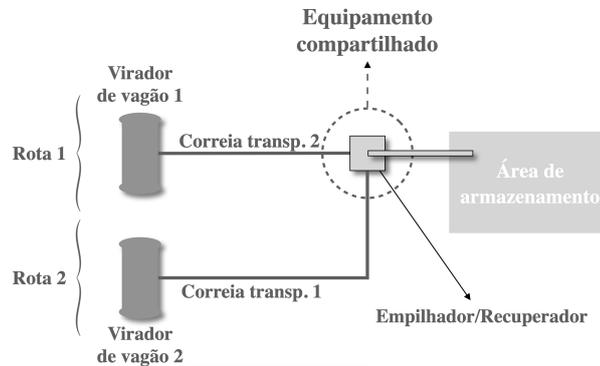


Figura 2 – Exemplos de rotas x que compartilham equipamento.

Os minérios de ferro procedentes das minas chegam no subsistema recepção através do sistema ferroviário. Na recepção, essas ofertas geradas pelos produtos são diretamente transferidas para uma área de estoque nos pátios. O subsistema Pátio é constituído de grandes áreas de armazenamento de produtos. Neste estudo, nós dividimos essas grandes áreas em áreas menores, nomeadas como subáreas. A capacidade limite de cada subárea é definida previamente, e depende do tipo do produto, produção das minas, demandas dos navios, entre outros fatores. Para evitar a contaminação dos produtos das pilhas, é mantido um espaço livre entre as subáreas. Na Figura 1 é possível visualizar os pátios divididos em subáreas e os espaços livres entre as subáreas. Cada subárea pode armazenar somente um produto por vez. Sempre que uma subárea ficar vazia e um novo diferente produto for atribuído na mesma, será considerado um custo de limpeza da subárea. O objetivo é fazer a melhor alocação de produto em cada subárea e, ao mesmo tempo, impedir que diferentes produtos sejam alocados na mesma subárea.

Quando um navio atraca em um berço, é gerada uma demanda que deve ser atendida. Os navios são carregados pelos equipamentos da rota y , que transportam os produtos alocados nas subáreas para os berços. Este estudo considera o *layout* discreto dos berços, no qual o cais é dividido em um conjunto finito de berços e apenas um navio pode ser designado para cada berço em um determinado tempo. Os berços podem ter diferentes tamanhos, desta forma cada navio deve atracar no berço correspondente ao seu tamanho. A chegada dos navios é considerada como estática, ou seja, todos os navios que devem ser atendidos já estarão disponíveis para carregar no início do horizonte de planejamento. No entanto, a sequência de atracação dos navios obtida na solução pode ser utilizada como planejamento de chegada dos navios. As atracações são limitadas pelos efeitos da maré. Com essa restrição, os navios só podem sair do berço em um período de maré alta, mesmo que o carregamento tenha sido concluído antes. A Figura 3 fornece um exemplo de alocação de berço com restrição de maré. Note que, após a saída de um determinado navio

do berço na maré alta, existe um tempo de desatracação e atracação do novo navio que vai atracar no mesmo berço. Como os navios chegam vazios no terminal, eles podem atracar em um período de maré baixa, como mostra a Figura 3. Os objetivos são alocar o navio no melhor berço e determinar a sequência mais eficiente de carregamento, considerando quantidade de produtos do estoque, comprimento do berço e efeitos da maré. A partir desses objetivos pretende-se reduzir os tempos de carregamento e custos com *demurrage*.

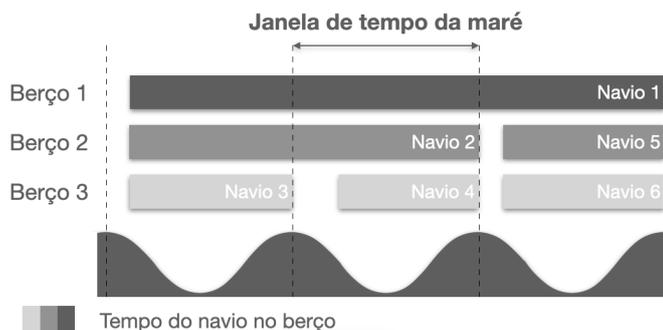


Figura 3 – Exemplo de alocação de berço com restrições de maré.

3.2 Modelos para o PIPSAPAB

Os modelos matemáticos descritos a seguir integram os problemas de planejamento, sequenciamento, alocação de pátio e alocação de berço de um terminal portuário graneleiro que manipula tipos de minério de ferro. O modelo matemático é baseado na formulação proposta por [Menezes, Mateus e Ravetti \(2017\)](#). Neste trabalho, consideramos apenas as rotas transportadoras x e y , enquanto que [Menezes, Mateus e Ravetti \(2017\)](#) consideram também uma terceira rota, que transporta os produtos diretamente do subsistema recepção para os berços. Além dessa mudança, o problema foi estendido com a adição do problema de alocação de berço e as restrições de sequenciamento das rotas foram reformuladas.

Embora este trabalho esteja focado em um tipo específico de terminal portuário, o problema, de uma maneira geral, bem como a sua formulação matemática, pode ser ajustado para tratar de diversos casos de terminais do tipo exportador. Alguns exemplos são: transporte de produtos agrícolas, transporte de carvão, entre outros.

O grande desafio desse modelo consiste em alocar e sequenciar as rotas para atender a oferta e a demanda, sequenciar e alocar cada navio e atribuir, para cada produto, uma subárea de forma integrada.

Neste trabalho, o horizonte de planejamento é discretizado em T períodos. Para considerar os efeitos da maré, os períodos devem possuir um tempo inferior a 12 horas. Esse tempo representa, aproximadamente, o intervalo entre as marés altas. O trabalho de [Butikov \(2002\)](#) apresenta uma explicação teórica sobre a periodicidade das ondas. Os navios entrarão no início de cada período e sairão apenas no final de cada período.

Os conjuntos (Tabela 1), parâmetros (Tabela 2) e variáveis de decisão (Tabela 3) do modelo são apresentados a seguir com uma breve descrição.

Tabela 1 – Definição dos conjuntos.

Conjunto	Descrição
$t \in T$	Conjunto de períodos.
$i \in I^t$	Conjunto de microperíodos.
$p \in P$	Conjunto de produtos.
$r \in R$	Conjunto de todas as rotas ($R = R^x \cup R^y$).
R^x	Subconjunto de rotas recepção/pátio.
R^y	Subconjunto de rotas pátio/pier.
E	Conjunto de pares de rotas que compartilham pelo menos um equipamento para transporte de produtos.
$s \in S$	Conjunto de subáreas.
R_s^x	Subconjunto de rotas x que chegam na subárea s .
R_s^y	Subconjunto de rotas y que partem da subárea s .
$m \in M$	Conjunto de equipamentos.
R_m^x	Subconjunto de rota x que utilizam o equipamento m .
R_m^y	Subconjunto de rota y que utilizam o equipamento m .
$b \in B$	Conjunto de berços de atracação
$n \in N$	Conjunto de navios.

Tabela 2 – Definição dos parâmetros.

Parâmetros	Descrição
O_{pt}	Oferta do produto p (em tonelada) no início do período t .
K_{np}	Quantidade de carga (em tonelada) do produto p para o navio n .
l_{pt}^s	Capacidade limite de armazenamento da subárea s para o produto p no período t .
b_m	Capacidade do equipamento m (em tonelada/hora).
j_t^m	Tempo disponível (em horas) para utilização do equipamento m no período t .
c^{rx}	Capacidade (em tonelada/hora) da rota $r \in R^x$.
c^{ry}	Capacidade (em tonelada/hora) da rota $r \in R^y$.
α_{pt}	Penalidade em não atender a oferta do produto p no período t .
β_{np}	Penalidade em não atender a demanda (carga) do produto p do navio n .
$\gamma_{pp't}^s$	Custo associado a troca do produto p pelo produto p' na subárea s o período t .
σ^r	Custo de manutenção pelo uso da rota r .
ϕ_n	Penalidade no tempo de permanência do navio n no terminal.
sb_b	Comprimento do berço b .
sv_n	Comprimento do navio n .
K_{max}	Constante igual ao valor do maior carregamento.
H_t	Duração máxima de cada período t .
μ_{max}	Duração máxima de cada microperíodo.
T_{max}	Constante igual ao número total de períodos.

Tabela 3 – Definição das variáveis de decisão.

Variáveis	Descrição
x_{pt}^r	Tempo (em horas) utilizado pelas rotas $r \in R^x$ para transportar o produto p da recepção para o pátio no período t .
y_{pt}^r	Tempo (em horas) utilizado pelas rotas $r \in R^y$ para transportar o produto p do pátio para o píer no período t .
d_{npt}^b	Quantidade do produto p carregada no navio n alocado no berço b no período t .
f_{pt}^s	Igual a 1 se o produto p está alocado na subárea s no período t , e 0 caso contrário.
$Sf_{pp't}^s$	Igual a 1 se o produto p é substituído pelo produto p' na subárea s período t , e 0 caso contrário.
e_{pt}^s	Quantidade do produto p armazenado na subárea s no período t .
IR_{pt}	Quantidade do produto p presente na recepção que não é atendida até final do período t .
IV_{np}	Quantidade do produto p que não foi carregada do navio n .
v_{nt}^b	Igual a 1 se o navio n está atracado no berço b no período t , e 0 caso contrário.
w_n^b	Igual a 1 se o berço b é atribuído para o navio n , e 0 caso contrário.
τ_n	Tempo de carregamento do navio n .
u_{pt}^r	Igual a 1 se o produto p usa a rota $r \in R$ no período t , e 0 caso contrário.
a_{pit}^r	Quando associada ao parâmetro μ_{max} , representa a duração do microperíodo $i \in I^t$ no período t da rota r que transporta o produto p .
q_{pit}^r	Igual a 1 se determinada rota r é ativada para transportar o produto p em microperíodo $i \in I^t$ no período t , e 0 caso contrário.
Δ_{it}	Duração total do microperíodo $i \in I^t$ no período t .

3.2.1 Formulação matemática PIPSAPAB-I

A formulação de otimização linear inteira para o PIPSAPAB-I é apresentado a abaixo.

- *Função objetivo*

$$\begin{aligned}
 \min f = & \sum_{p \in P} \sum_{t \in T} \alpha_{pt} IR_{pt} + \sum_{n \in N} \sum_{p \in P} \beta_{np} IV_{np} + \sum_{s \in S} \sum_{p \in P} \sum_{p' \in P} \sum_{t \in T} \gamma_{pt}^s S f_{pp't}^s \\
 & + \sum_{p \in P} \sum_{t \in T} \sum_{r \in R^x} \sigma^r (c^{rx} x_{pt}^r) + \sum_{p \in P} \sum_{t \in T} \sum_{r \in R^y} \sigma^r (c^{ry} y_{pt}^r) + \sum_{n \in N} \phi_n \tau_n
 \end{aligned} \tag{1}$$

A função objetivo busca minimizar os custos e as penalidades relativas ao não atendimento das ofertas e carregamentos, alocação de produtos nos pátios, utilização das rotas e duração do carregamento dos navios. O primeiro termo expressa as penalidades de não atendimento das ofertas que chegam na recepção. O segundo termo expressa as penalidades em não atender as demandas dos navios. O terceiro termo representa os custos

de limpeza na troca de produtos nas subáreas. O quarto e o quinto termos são referentes aos custos de utilização das rotas que transportam os produtos. O último termo expressa as penalidades sobre o tempo de carregamento dos navios.

- *Restrições*

$$\sum_{r \in R^x} c^{rx} x_{pt}^r - IR_{p(t-1)} + IR_{pt} = O_{pt} \quad \forall p \in P, \forall t \in T \quad (2)$$

As restrições (2) controlam o atendimento da oferta no subsistema recepção. Estas restrições garantem que o fluxo de produtos da recepção para o pátio, mais a oferta não atendida, é igual a oferta do produto p no período t . Note que a oferta não atendida no período anterior é transferida para o período seguinte ($IR_{p(t-1)}$). As ofertas não atendidas no período zero (IR_{p0}) são dados de entrada para o problema.

$$\sum_{r \in R^y} c^{ry} y_{pt}^r = \sum_{n \in N} d_{npt}^b \quad \forall b \in B, \forall p \in P, \forall t \in T \quad (3)$$

As restrições (3) controlam o carregamento dos navios no subsistema píer. Estas restrições garantem que o fluxo de produtos que sai do pátio e chega aos berços é igual ao somatório dos montantes do produto p que carrega o navio n alocado no berço b no período t . Note que o conjunto de rotas y é dividido em conjunto menor de rotas que chegam em apenas um determinado berço b . Com isso, o somatório $\sum_{r \in R^y} c^{ry} y_{pt}^r$ deve apresentar apenas rotas que chegam no berço b .

$$\sum_{b \in B} \sum_{t \in T} d_{npt}^b + IV_{np} = K_{np} \quad \forall n \in N, \forall p \in P \quad (4)$$

As restrições (4) controlam o atendimento da demanda dos navios. O somatório de carregamentos efetuadas pelas rotas y mais a carga não atendida é igual a carga do produto p demanda pelo navio n .

$$\sum_{n \in N} v_{nt}^b = 1 \quad \forall b \in B, \forall t \in T \quad (5)$$

As restrições (5) impõem que a partir do momento em que um navio n for alocado em um berço b , este não pode ser mais utilizado por nenhum outro navio no período t .

$$\sum_{p \in P} d_{npt}^b \leq K_{max} v_{nt}^b \quad \forall b \in B, \forall n \in N, \forall t \in T \quad (6)$$

As restrições (6) asseguram que um navio n pode ser carregado somente quando estiver atracado no berço b . As restrições (3), (5) e (6) são complementares e garantem

que os navios sejam carregados apenas dentro dos períodos que estiverem alocados em um berço. Note que, no somatório $\sum_{b \in B} \sum_{n \in N} d_{npt}^b$, somente as variáveis correspondentes aos navios que estiverem alocados nos berços no período t serão diferente de zero (tendo valores iguais à quantidade de produto que está carregando do período), as demais serão iguais a zero (representando que os navios não estão sendo carregadas por ele não estar alocado em nenhum berço no período t). Por exemplo, se o navio 1 é carregado no berço 1 no período 1, logo $d_{1p1}^1 \geq 0$, $d_{1p1}^{b \neq 1} = 0$.

$$\sum_{m=1}^{t-1} v_{nm}^b - v_{n(t-1)}^b + v_{nt}^b \leq t \quad \forall b \in B, \forall n \in N, \forall t \in T : t > 1 \quad (7)$$

As restrições (7) garantem a não interrupção na operação de carregamento dos navios. Desse modo, os navios começam a ser carregados no início do período que atracam e saem do berço somente quando o carregamento finalizar. Estas restrições são baseadas na formulação proposta por [Barros et al. \(2011\)](#).

$$\sum_{t \in T} v_{nt}^b = 0 \quad \forall b \in B, \forall n \in N : (s_b < sv_n) \quad (8)$$

As restrições (8) impedem que os navios com comprimento superior do berço sejam atracados. Se $\sum_{t \in T} v_{nt}^b = 0$, o navio n não pode atracar no berço b em nenhum período. Estas restrições são baseadas na formulação proposta por [Unsal e Oğuz \(2019\)](#).

$$\sum_{b \in B} \sum_{t \in T} v_{nt}^b = \tau_n \quad \forall n \in N \quad (9)$$

As restrições (9) são encarregadas em contabilizar todos os períodos utilizados para carregar cada navio n . Estas restrições estão associadas ao último termo da função objetivo.

$$\sum_{t \in T} v_{nt}^b \leq T_{max} w_n^b \quad \forall b \in B, \forall n \in N \quad (10)$$

$$\sum_{b \in B} w_n^b = 1 \quad \forall n \in N \quad (11)$$

As restrições (10) e (11) controlam a alocação de berços para os navios. As restrições (11) impõem que, se um navio n for alocado no berço b , ele não pode ser mais alocado em nenhum outro berço. Nas restrições (10), se $w_n^b = 1$, possibilita que as variáveis v_{nt}^b sejam diferente de zero, logo permitindo que um navio n seja atribuído ao berço b em qualquer período. Devido à formulação das restrições associadas à alocação de berço, a quantidade de navios deve ser superior à quantidade de berços.

$$e_{p(t+1)}^s = e_{pt}^s + \sum_{r \in R_s^x} c^{rx} x_{pt}^r - \sum_{r \in R_s^y} c^{ry} y_{pt}^r \quad \forall s \in S, \forall p \in P, \forall t \in T \quad (12)$$

$$e_{pt}^s \leq l_{pt}^s \quad \forall s \in S, \forall p \in P, \forall t \in T \quad (13)$$

As restrições (12) são de balanceamento de estoque: o estoque do produto p na subárea s no período $t + 1$ é igual ao estoque no período t mais o fluxo de produtos que entra (rotas x) menos o fluxo de produtos que sai (rotas y). O estoque inicial das subáreas (e_{p1}^s) é dado como entrada para o problema. As restrições (13) limitam a capacidade de armazenamento de cada subárea.

$$\sum_{p \in P} f_{pt}^s = 1 \quad \forall s \in S, \forall t \in T \quad (14)$$

$$l_{pt}^s f_{pt}^s - e_{pt}^s \geq 0 \quad \forall s \in S, \forall t \in T \quad (15)$$

$$l_{pt}^s f_{pt}^s - \sum_{r \in R_s^x} x_{pt}^r \geq 0 \quad \forall s \in S, \forall t \in T \quad (16)$$

As restrições (14)-(16) controlam a alocação dos produtos nas subáreas. As restrições (14) garantem que apenas um produto p pode ser alocado em uma subárea s no período t . As restrições (15) relacionam as variáveis f_{pt}^s e e_{pt}^s , permitindo que o produto alocado na subárea seja armazenado. As restrições (16) são responsáveis por viabilizar a entrada do produto alocado na determinada subárea, através do fluxo de produtos (rotas x).

$$S f_{pp't}^s \geq f_{p(t-1)}^s + f_{p't}^s - 1 \quad \forall s \in S, \forall t \in T, \forall p \in P, \forall p' \in P, p \neq p' \quad (17)$$

As restrições (17) controlam a troca de produtos nas subáreas. Se $S f_{pp't}^s = 1$, o produto p foi substituído pelo p' na subárea s no período t . Estas restrições estão relacionadas à manutenção/limpeza de uma subárea, e apresenta um termo associado na função objetivo do modelo.

$$\sum_{p \in P} \left(\sum_{r \in R_m^x} c^{rx} x_{pt}^r + \sum_{r \in R_m^y} c^{ry} y_{pt}^r \right) \leq j_t^m b^m \quad \forall m \in M, \forall t \in T \quad (18)$$

As restrições (18) garantem que a capacidade de nenhum equipamento seja excedida. Estas restrições permitem que rotas que compartilham equipamentos sejam utilizadas em um mesmo período e ao mesmo tempo. Para complementar estas restrições e não permitir o compartilhamento simultâneo de rotas conflitantes, são incluídas as restrições de sequenciamento.

$$x_{pt}^r \leq H_t u_{pt}^r \quad \forall p \in P, \forall t \in T, \forall r \in R^x \quad (19)$$

$$y_{pt}^r \leq H_t u_{pt}^r \quad \forall p \in P, \forall t \in T, \forall r \in R^y \quad (20)$$

Se a rota r é utilizada no período t para transportar o produto p ($u_{pt}^r = 1$), as restrições (19) e (20) asseguram a sua disponibilidade e capacidade (medida em horas).

$$x_{pt}^r \leq \sum_{i \in I^t} a_{pit}^r \mu_{max} \quad \forall p \in P, \forall t \in T, \forall r \in R^x \quad (21)$$

$$y_{pt}^r \leq \sum_{i \in I^t} a_{pit}^r \mu_{max} \quad \forall p \in P, \forall t \in T, \forall r \in R^y \quad (22)$$

$$a_{pit}^r \leq q_{pit}^r \quad \forall p \in P, \forall i \in I^t, \forall t \in T, \forall r \in R \quad (23)$$

$$q_{pit}^r + q_{pit}^{r'} \leq 1 \quad \forall p \in P, \forall i \in I^t, \forall t \in T, \forall (r, r') \in E \quad (24)$$

$$a_{pit}^r \mu_{max} \leq \Delta_{it} \quad \forall p \in P, \forall i \in I^t, \forall t \in T, \forall r \in R \quad (25)$$

$$\sum_{i \in I^t} \Delta_{it} \leq H_t \quad \forall t \in T \quad (26)$$

As restrições (21)-(26) estão associadas ao sequenciamento. Para o sequenciamento, cada período t é dividido em uma quantidade fixa de microperíodos. O número de microperíodos é definido como entrada do problema. A duração máxima dos microperíodo (μ_{max}) é igual ao maior tempo (em horas) para utilização do equipamento, ou seja, maior valor do parâmetro j_t^m . O parâmetro H_t estabelece a duração máxima de cada período t . Cada microperíodo irá conter apenas as rotas que podem ser executadas simultaneamente, ou seja, que não possuem conflito em seus equipamentos. Os somatórios das restrições (21) e (22) referem-se aos microperíodos disponíveis para cada rota. As variáveis x_{pt}^r e y_{pt}^r podem ser interpretadas como atividades de transporte, uma vez que os seus valores representam as horas necessárias para transportar os produtos entre os subsistemas. No entanto, para permitir o transporte dos produtos é necessário alocar um número de microperíodos para cada uma destas rotas.

As restrições (23) e (24) são responsáveis por ativar apenas os microperíodos que não apresentam conflito entre os equipamentos. Nas restrições (23), quando o microperíodo é ativado ($q_{pit}^r = 1$), a variável a_{pit}^r pode possuir um valor entre 0 e 1, e conseqüentemente permitindo o microperíodo ter um tempo variável. Note que a duração de microperíodo é dado pelo produto das variáveis a_{pit}^r e o parâmetro μ_{max} .

Juntas as restrições (25) e (26) são responsáveis por limitar a duração dos períodos. As restrições (25) determinam a duração total de cada microperíodo i do período t (Δ_{it}). Enquanto as restrições (26) asseguram que o somatório de todos os microperíodos pertencentes ao período t deve ser menor que a duração máxima do período (H_t). Note que os microperíodos são contínuos, ou seja, ao final de um microperíodo inicia-se o microperíodo seguinte.

$$d_{npt}^b, x_{pt}^r, y_{pt}^r, e_{pt}^s, IR_{pt}, IV_{np}, \tau_n, \Delta_{it} \geq 0 \quad (27)$$

$$\forall r \in R, \forall b \in B, \forall n \in N, \forall s \in S, \forall p \in P, \forall i \in I^t, \forall t \in T$$

$$f_{pt}^s, v_{nt}^b, w_n^b, u_{pt}^r, q_{pit}^r \in \{0, 1\} \quad (28)$$

$$\forall r \in R, \forall b \in B, \forall n \in N, \forall s \in S, \forall p \in P, \forall t \in T, \forall i \in I^t$$

$$0 \leq S_{pp't}^s \leq 1 \quad \forall s \in S, \forall p \in P, \forall p' \in P, p \neq p', \forall t \in T \quad (29)$$

$$0 \leq a_{pit}^r \leq 1 \quad \forall r \in R, \forall p \in P, \forall i \in I^t, \forall t \in T \quad (30)$$

As restrições (27)-(30) são de não negatividade e integralidade das variáveis do modelo.

Portanto, as expressões (1)-(30) definem PIPSAPAB-I.

3.2.2 Formulação matemática PIPSAPAB-II

Além das restrições (21)-(26) denotadas anteriormente, existem outras formulações alternativas podem ser investigadas para o problema de sequenciamento. A formulação proposta nesta seção apresenta microperíodos com duração de tempo menor, e a alocação consecutiva de microperíodos ativados para uma mesma rota. As restrições de sequenciamento proposto nesta seção consistem em dividir o período t em microperíodos $i \in I^t$, e que I^t representa o número total de microperíodos disponíveis para o período t , sendo definido como entrada do problema. A duração de cada microperíodo é fixa e dada pela seguinte expressão: $\mu_{it} = H_t/I_t$, sendo H_t o parâmetro que estabelece a duração máxima de cada período t . Cada microperíodo (μ_{it}) irá conter apenas as rotas que podem ser ativadas ao mesmo tempo, ou seja, apenas rotas que não possuem conflito em seus equipamentos. Esta condição é garantida pelas restrições (33).

$$x_{pt}^r \leq \sum_{i \in I^t} q_{pit}^r \mu_{it} \quad \forall p \in P, \forall t \in T, \forall r \in R^x \quad (31)$$

$$y_{pt}^r \leq \sum_{i \in I^t} q_{pit}^r \mu_{it} \quad \forall p \in P, \forall t \in T, \forall r \in R^y \quad (32)$$

$$q_{pit}^r + q_{pit}^{r'} \leq 1 \quad , \forall i \in I^t \forall p \in P, \forall (r, r' \in E) \forall t \in T \quad (33)$$

$$\sum_{k=1}^{i-1} q_{pkt}^r - q_{p(i-1)t}^r + q_{pit}^r \leq i \quad \forall r \in R, \forall p \in P, \forall t \in T, \forall i \in I^t : i > 1 \quad (34)$$

Como foram alteradas apenas as restrições de sequenciamento, as variáveis x_{pt}^r e y_{pt}^r continuam representando as horas utilizadas para transportar os produtos entre os subsistemas. Da mesma forma que em (21) e (22), os somatórios das restrições (31) e (32) representam os microperíodos disponíveis para cada rota, e que devem ser atribuídos de acordo com a duração das variáveis x_{pt}^r e y_{pt}^r .

As restrições (34) garantem que os microperíodos serão atribuídas de maneira consecutivas nas rotas ativas, dessa forma impedindo a interrupção das tarefas. Com essas

restrições, problemas com preempção são evitados. As restrições (34) são semelhantes as restrições (7), pois ambas, são capazes de eliminar os 1's não consecutivas das variáveis q_{pit}^r e v_{nt}^b , respectivamente. Da mesma forma que as restrições (7), as restrições (34) são baseadas na formulação proposta em Barros et al. (2011).

As Figuras 4 e 5 mostram as diferenças entre as restrições de sequenciamento (21)-(26) e (31)-(34). Essas figuras ilustram a alocação dos microperíodos na solução de ambas formulações.

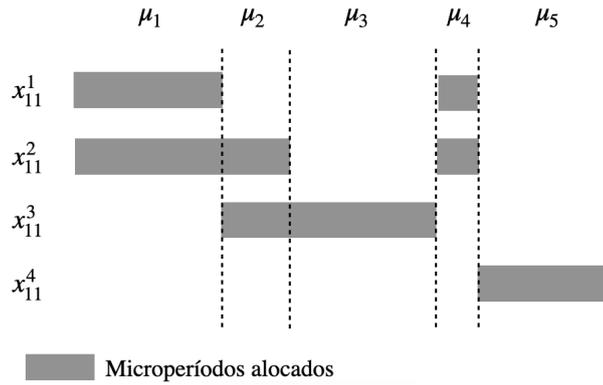


Figura 4 – Exemplo de solução com preempção dos microperíodos.

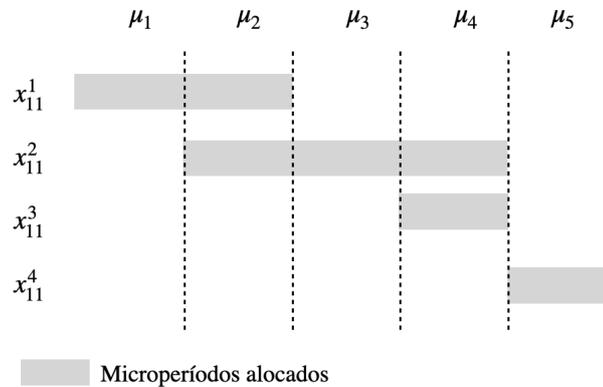


Figura 5 – Exemplo de solução sem preempção dos microperíodos.

A comparação das duas figuras destaca as diferenças entre dois conjuntos de restrições. Na Figura 4, verifica-se que os microperíodos alocados para uma determinada rota não são alocados de maneira consecutiva; como exemplo, a rota 1 (x_{11}^1) possui alocado os microperíodos 1 e 4. Na Figura 5, os microperíodos são alocados consecutivamente para uma determinada rota ativada, como exemplo; a rota 1 (x_{11}^1) apresenta os microperíodos 1 e 2 alocados. Esta configuração dos microperíodos representa que uma determinada rota não terá nenhuma interrupção no transporte até que todo o lote seja finalizado. Nas duas figuras também é possível visualizar a duração variável dos microperíodos na formulação apresentada na seção, e a duração fixa dos microperíodos na formulação desta seção.

As restrições de sequenciamento incorporadas na formulação PIPSAPAB-II originam um novo problema. Por ser um modelo integrado complexo e com grande número de restrições, a modificação ou adição de algumas restrições pode afetar outras, e assim exigir adição de novas restrições para manter a coerência da formulação com o problema definido. Com as alterações na parte do sequenciamento, as restrições (12) apresentaram um falha, pois as variações do estoque que são controladas pelas variáveis e_{pt}^s e $e_{p(t+1)}^s$ podem vir a ficar igual a zero, enquanto que as rotas x e y transportam exatamente a mesma quantidade de produto, o que é errado, pois as rotas y devem transportar somente os produtos que estão no estoque no período. No ponto de vista de um terminal, esse erro significaria que ao mesmo tempo que o produto chega no subsistema de Recepção, ele está saindo de uma determinada subárea. Uma justificativa para isso está associada às rotas com menores custos. Caso chegue na recepção a oferta de um determinado produto, e no mesmo período tem-se um navio atracado que demanda desse mesmo produto, e ainda, existe a possibilidade desse produto ser transportado diretamente por uma rota com o custo baixo, isso será incorporado na solução do problema, pois as restrições (12) permitem essa falha. Portanto, para restaurar a concordância da formulação e reforçar as restrições (12), é proposta a seguinte restrição.

$$e_{pt}^s \geq \sum_{r \in R_s^y} c^{ry} y_{pt}^r \quad \forall s \in S, \forall p \in P, \forall t \in T \quad (35)$$

As restrições (35) garantem que as rotas y transportaram somente os produtos que estiverem no estoque no início do período. Desse modo, impedindo a passagem direta e incorreta do fluxo de produtos entre as rotas x e y .

Para flexibilizar a formulação, as restrições (5) e (14) da formulação PIPSAPAB-I foram alteradas.

$$\sum_{n \in (N \cup 0)} v_{nt}^b = 1 \quad \forall b \in B, \forall t \in T \quad (36)$$

As restrições (36) continuam impedindo que um berço atende mais de um navio em um período t , no entanto permitem que o berço permaneça vazio quando o navio 0 ($n = 0$) estiver alocado no período t .

$$\sum_{p \in (P \cup 0)} f_{pt}^s = 1 \quad , \forall s \in S, \forall t \in T \quad (37)$$

Da mesma forma que as (14), as restrições (37) também alocam somente um produto em cada subárea por vez, porém possibilitam que as subáreas permaneçam vazias quando o produto 0 ($p = 0$) for atribuído no período t .

$$d_{npt}^b, x_{pt}^r, y_{pt}^r, e_{pt}^s, IR_{pt}, IV_{np}, \tau_n \geq 0 \quad (38)$$

$$\forall r \in R, \forall b \in B, \forall n \in N, \forall s \in S, \forall p \in P, \forall t \in T$$

$$f_{pt}^s, v_{nt}^b, w_n^b, u_{pt}^r, q_{pit}^r \in \{0, 1\} \quad (39)$$

$$\forall r \in R, \forall b \in B, \forall n \in N, \forall s \in S, \forall p \in P, \forall t \in T, \forall i \in I^t$$

$$0 \leq S_{pp't}^s \leq 1 \quad (40)$$

$$\forall r \in R, \forall s \in S, \forall p \in P, \forall p' \in P, p \neq p', \forall i \in I^t, \forall t \in T$$

As restrições (38)-(40) são de não negatividade e integralidade das variáveis da formulação alternativa.

3.2.2.1 Restrições adicionais

Além das restrições apresentadas anteriormente, foram consideradas as seguintes restrições com o propósito de deixar a formulação matemática mais justa.

$$\sum_{b \in B} \sum_{t \in T} v_{nt}^b \geq 1 \quad , \forall n \in N \quad (41)$$

As restrições (41) foram adicionadas para assegurar que todos os navios serão atendidos em ao menos um período t . Como um dos objetivos da formulação é atender a toda a fila de navios e o não atendimento de um navio torna o valor da função objetivo muito maior, a adição dessas restrições reduz o espaço da solução impondo que todos os navios devem ser atendidos ao menos em um período. Essas restrições auxiliam o algoritmo heurístico, que será apresentado a seguir, a obter uma solução inteira das variáveis v_{nt}^b .

$$\sum_{s \in S} \sum_{t \in T} f_{pt}^s \geq 1 \quad , \forall p \in P \quad (42)$$

As restrições (42) foram adicionadas para assegurar que todos os produtos serão alocados em ao menos um período t . Como o não atendimento de uma oferta é muito custoso para o valor de f , o incremento dessas restrições auxilia alocar todos os produtos em alguma subárea, logo reduzindo a tendência de soluções com produtos não atendidos no subsistema de recepção.

Portanto, as expressões (1)-(4), (6)-(13), (15)-(20) e (31)-(42) definem o PIPSAPAB-II.

3.2.3 Considerações finais

As formulações matemáticas PIPSAPAB-I e PIPSAPAB-II foram projetadas para modelar o mesmo problema integrado de planejamento, sequenciamento, alocação de pátio e alocação de berço. Logo, as formulações apresentam semelhanças, mas também

diferenças. A principal diferença entre elas são as restrições de sequenciamento. A formulação PIPSAPAB-II considera um sequenciamento com microperíodos fixos e atribuídos de maneira consecutiva para uma determinada rota ativa. Enquanto que, a formulação PIPSAPAB-I apresenta um sequenciamento com microperíodos variáveis e atribuídos de forma não consecutiva para uma determinada rota ativa. Além disso, os microperíodos fixos da formulação PIPSAPAB-II apresentam uma duração menor que em relação à formulação PIPSAPAB-I. Além dessas, outras diferenças entre as formulações estão associadas a alocação de produtos nas subáreas e a atribuição de embarcações nos berços. As restrições (37) e (36) da formulação PIPSAPAB-II permitem que as subáreas fiquem sem produtos alocados e os berços permaneçam momentaneamente vazios, enquanto que as restrições (14) e (5) da formulação PIPSAPAB-I não permitem isso. Os capítulos seguintes apresentam uma abordagem de solução para resolver cada formulação. O Capítulo 4 apresenta uma abordagem de solução para a formulação PIPSAPAB-I, enquanto que o Capítulo 5 para a formulação PIPSAPAB-II.

Capítulo 4

Abordagem de solução para o PIPSAPAB-I

Este capítulo descreve uma abordagem de solução alternativa para a formulação matemática PIPSAPAB-I com uso da técnica de Geração de Colunas.

4.1 Decomposição de Dantzig-Wolfe e Geração de Colunas

A decomposição de Dantzig-Wolfe foi introduzida em [Dantzig e Wolfe \(1960\)](#). A ideia básica é dividir o problema em subproblemas menores que podem ser resolvidos de forma independente no nível inferior, em que as soluções (colunas) são então coordenadas e combinadas no nível superior, denominado problema mestre. O método de solução chamado de geração de colunas, permite lidar com um grande número de variáveis, uma vez que apenas colunas promissoras para solução são adicionadas no problema mestre e a decisão é baseada no custo reduzido das colunas, que são precificadas no subproblema (*pricing*).

Nas últimas décadas, a decomposição de Dantzig-Wolfe e a geração de colunas para problemas inteiros foram amplamente estudadas e aplicadas em uma variedade de problemas. Uma lista com algumas das mais recentes aplicações e referências é exposta na [Tabela 4](#).

Aplicação	Referências
<i>bin packing</i>	Casazza e Ceselli (2016), Zhang et al. (2017), Mahvash, Awasthi e Chauhan (2018)
<i>roteamento de veículos</i>	Kulkarni et al. (2018), Gschwind et al. (2018), Tchoupo et al. (2018), Faiz, Vogiatzis e Noor-E-Alam (2019), Taş (2020), Ibrahim et al. (2020), Himmich, Hallaoui e Soumis (2020), Munari e Savelsbergh (2020)
<i>sequenciamento</i>	Restrepo, Gendron e Rousseau (2018), Nguyen et al. (2019), Alfieri et al. (2019), Zhang, Dridi e El Moudni (2020), Pei et al. (2020), Ozturk (2020)
<i>operações aéreas</i>	Koháni e Janáček (2017), Janacek et al. (2017), Frey, Kolisch e Artigues (2017), Liang et al. (2018)
<i>transporte marítimo</i>	Menezes, Mateus e Ravetti (2017), Wang et al. (2018), Zhen et al. (2020)

Tabela 4 – Algumas trabalhos de decomposição de Dantzig-Wolfe e Geração de Colunas.

4.2 Formulação baseada na geração de colunas

A formulação matemática PIPSAPAB-I apresentada anteriormente é extremamente complexo e desafiador. O grande número de restrições e variáveis inteiras, bem como a ampla quantidade de possíveis soluções para o problema, especialmente para o sequenciamento dos equipamentos e rotas, tornam totalmente inviável o uso de *solvers* comerciais, como CPLEX, quando as instâncias do problema abordam situações reais. Para tentar contornar essas limitações, o presente trabalho aplica método de Decomposição de Dantzig-Wolfe, em que o MIP apresentado é decomposto em um problema mestre linear restrito (PMLR) e em um subproblema de *pricing*.

4.2.1 Problema Mestre Linear Restrito

Na formulação compacta PIPSAPAB-I, o sequenciamento de equipamentos e rotas é garantido pelas restrições (19)-(26). Em especial estas restrições requerem um grande consumo de memória e processamento, e estabelecem dificuldades para solucionar o problema integrado. Com a decomposição do modelo, o PMLR equivalerá ao planejamento da produção, alocação de berço e alocação de pátio, e o subproblema de precificação representará o sequenciamento dos equipamentos e rotas e será encarregado de gerar colunas para o PMLR. Um dos objetivos desse método é reduzir a quantidade de variáveis para somente aquelas que forem mais convenientes à solução do problema. Para isto, novamente cada período t será dividido em microperíodos. Para manter a consistência entre as duas formulações, a duração dos microperíodos é limitada pelo parâmetro μ_{max} . A seguir são apresentados os novos parâmetros e variáveis utilizados na formulação do PMLR e no subproblema.

Tabela 5 – Variáveis e parâmetros adicionais utilizados para a geração de colunas.

Variáveis	Descrição
μ_{it}	Duração (em horas) do microperíodo $i \in I^t$ no período t .
Parâmetros	Descrição
ψ_{pit}^r	Igual a 1 se a rota r for usada no microperíodo i no período t (parâmetro associado ao valor da variável q_{pit}^r resultante da solução do subproblema), e 0 caso contrário

No contexto de decomposição, as restrições (19) e (20) podem ser reescritas da seguinte maneira:

$$x_{pt}^r \leq \sum_{i \in I^t} \psi_{pit}^r \mu_{it} \quad \forall r \in R^x, \forall p \in P, \forall t \in T \quad (43)$$

$$y_{pt}^r \leq \sum_{i \in I^t} \psi_{pit}^r \mu_{it} \quad \forall r \in R^y, \forall p \in P, \forall t \in T \quad (44)$$

$$\sum_{i \in I^t} \mu_{it} \leq H_t \quad \forall t \in T \quad (45)$$

$$0 \leq \mu_{it} \leq \mu_{max} \quad \forall i \in I^t, \forall t \in T \quad (46)$$

As variáveis μ_{it} representam a duração do microperíodo i no período t para cada produto alocado a r . Deste modo, as durações das rotas x_{pt}^r e y_{pt}^r estão limitadas aos microperíodos alocados para cada uma.

Com a decomposição do modelo, o PMLR é definido pelas restrições de planejamento da produção, alocação de berços e alocação de pátios da formulação PIPSAPAB-I, e também pelas novas restrições (43)-(46) (que substituem as restrições (19) e (20)). O subproblema será encarregado de gerar as colunas relativas aos microperíodos, e assim garantir o sequenciamento das rotas para cada período, atendendo as restrições (21)-(26) da formulação compacta. A seguir é apresentada a formulação do PMLR.

$$\begin{aligned} \min f = & \sum_{p \in P} \sum_{t \in T} \alpha_{pt} I R_{pt} + \sum_{n \in N} \sum_{p \in P} \beta_{np} I V_{np} + \sum_{s \in S} \sum_{p \in P} \sum_{p' \in P} \sum_{t \in T} \gamma_{pt}^s S f_s^{pp't} \\ & + \sum_{p \in P} \sum_{t \in T} \sum_{r \in R^x} \sigma^r(c^{rx} x_{pt}^r) + \sum_{p \in P} \sum_{t \in T} \sum_{r \in R^y} \sigma^r(c^{ry} y_{pt}^r) + \sum_{n \in N} \phi_n \tau_n \end{aligned} \quad (47)$$

sujeito a

restrições (2)-(18)

$$x_{pt}^r \leq \sum_{i \in I^t} \psi_{pit}^r \mu_{it} \quad \forall r \in R^x, \forall p \in P, \forall t \in T \quad (48)$$

$$y_{pt}^r \leq \sum_{i \in I^t} \psi_{pit}^r \mu_{it} \quad \forall r \in R^y, \forall p \in P, \forall t \in T \quad (49)$$

$$\sum_{i \in I^t} \mu_{it} \leq H_t \quad \forall t \in T \quad (50)$$

$$0 \leq \mu_{it} \leq \mu_{max} \quad \forall i \in I^t, \forall t \in T \quad (51)$$

$$w_n^b, v_{nt}^b, f_{pt}^s \geq 0 \quad (52)$$

$$\forall b \in B, \forall n \in N, \forall s \in S, \forall p \in P, \forall t \in T$$

$$0 \leq S_{pp't}^s \leq 1 \quad (53)$$

$$\forall s \in S, \forall p \in P, \forall p' \in P, p \neq p', \forall t \in T$$

$$d_{npt}^b, x_{pt}^r, y_{pt}^r, e_{pt}^s, IR_{pt}, IV_{np}, \tau_n, \Delta_{it} \geq 0 \quad (54)$$

$$\forall r \in R, \forall b \in B, \forall n \in N, \forall s \in S, \forall p \in P, \forall i \in I^t, \forall t \in T$$

A função objetivo permanece inalterável com relação à apresentada na formulação PIPSAPAB-I. Semelhante às restrições (21) e (22), as restrições (48) e (49) garantem que os microperíodos alocadas para cada rota sejam iguais ou superiores ao tempo necessário para transportar os produtos, uma vez que os microperíodos alocados não permitem o conflito dos equipamentos. Essa restrição será imposta resolvendo o subproblema precificação, que fornece colunas (microperíodos) para o PMLR. As restrições (50) limitam o número de microperíodos que podem estar ativos em um período, como também, da mesma forma que as restrições (26), impedem que a duração do período t seja maior que o máximo de H_t horas. As restrições (46) delimitam a duração máxima de cada microperíodo. As restrições (52) são referentes às variáveis binárias relaxadas com a decomposição do modelo compacto. As restrições (54) e (53) são referentes aos limites das variáveis.

4.2.2 Subproblema de precificação

Associando as variáveis duais π_{pt}^r às restrições (43) e (44), as variáveis duais η_t nas restrições (45), e como os custos associados às variáveis μ_{it} na função objetivo são nulos, o custo reduzido associado a uma variável μ_{it} no período t é dado por:

$$- \sum_{p \in P} \sum_{r \in R} \pi_{pt}^r \psi_{pit}^r + \eta_t \quad \forall t \in T$$

Caso o valor do custo reduzido seja menor que zero, a variável (microperíodo) μ_{it} deve ser adicionada na nova coluna que integrará no problema mestre. Caso o valor do custo reduzido for maior que zero, a condição de otimalidade está atendida. A partir dessas considerações, o subproblema busca gerar uma coluna com as soluções da variável q_{pit}^r associada ao microperíodo μ_{it} tal que:

$$\mathbf{max} \sum_{p \in P} \sum_{r \in R} \pi_{pt}^r q_{pit}^r - \eta_t$$

Portanto, o subproblema para cada microperíodo i no período t , é definido como:

$$\mathbf{max} \sum_{p \in P} \sum_{r \in R} \pi_{pt}^r a_{pit}^r - \eta_t \quad (55)$$

subj. a

$$q_{pit}^r + q_{pit}^{r'} \leq 1 \quad \forall p \in P, \forall (r, r') \in E, \forall t \in T \quad (56)$$

$$q_{pit}^r \in \{0, 1\} \quad \forall r \in R, \forall p \in P, \forall t \in T, \forall r \in R \quad (57)$$

Em cada iteração da geração de coluna é solucionado um subproblema para cada período t . Cada subproblema tem o objetivo de gerar uma coluna de microperíodos que contenha as melhores rotas que podem ser ativadas simultaneamente, isto é, respeita a condição de que o mesmo equipamento seja não utilizado por mais de uma rota. As colunas são geradas até que o custo reduzido seja menor que zero.

A função objetivo busca maximizar o número de rotas não conflitantes ativadas no microperíodo i no período t , segundo os valores de π_{pt}^r . As restrições (56) garantem o sequenciamento das rotas sem que tenha conflito entre as rotas, assim ativando somente os microperíodos que podem ser executados simultaneamente. Se $q_{pit}^r = 1$, a rota r que transporta o produto p pode ser alocada no microperíodo i no período t .

Como cada subproblema é solucionado separadamente para cada microperíodo i de período t e a complexidade do problema é significativamente reduzida. Além disso, se o valor da variável η_t na função objetivo for zero, a formulação matemática do problema do conjunto independente de peso máximo (PCIPM) e do subproblema *pricing* são idênticas. Os subproblemas são resolvidos de maneira exata pelo *solver* CPLEX. A Figura 6 ilustra a estratégia de decomposição desenvolvida para solucionar o problema integrado estudado.

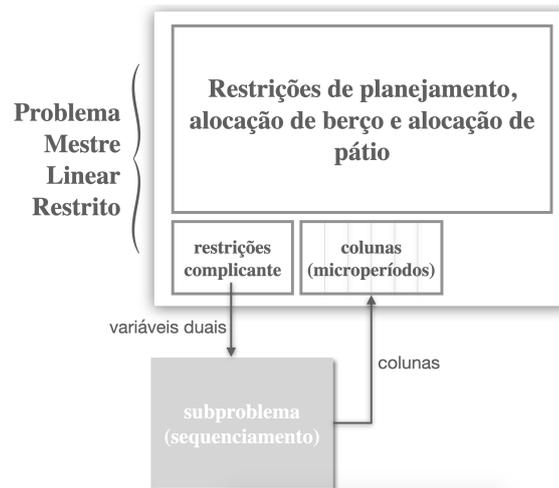


Figura 6 – Decomposição do modelo.

4.2.3 Heurística Baseada na Geração de Colunas

O procedimento de geração de colunas proposto apenas soluciona a relaxação linear do PMLR. A solução obtida após a convergência deste procedimento não é inteira. Apenas a

conversão para inteiro das variáveis relaxadas do PMLR e sua resolução usando o *solver* podem não fornecer uma solução de qualidade para o problema integrado. Novas colunas que poderiam melhorar a solução do PMLR não são inseridas nesse procedimento. Com o objetivo de fornecer soluções inteiras de boa qualidade, propomos uma heurística baseada em geração de colunas (CGBH). Essa heurística é dividida em três partes, na qual, em cada parte, um conjunto de variáveis são fixadas. A Figura 7 exibe um fluxograma com as etapas da CGBH.

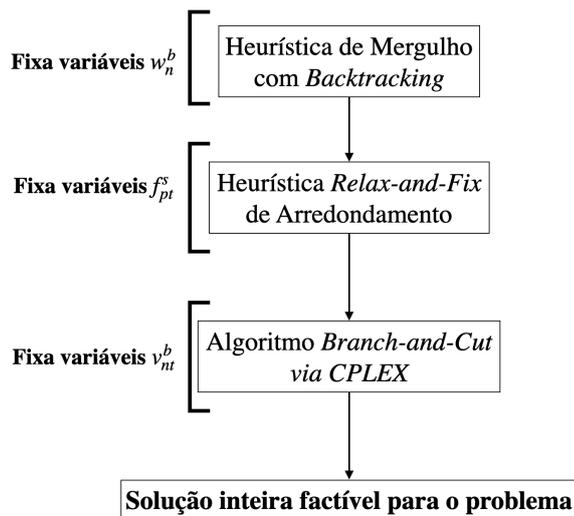


Figura 7 – Fluxograma da heurística baseada em geração de colunas.

4.2.3.1 Heurística de Mergulho com *Backtracking*

Com a decomposição do modelo, as variáveis binárias w_n^b , v_{nt}^b e f_{pt}^s são relaxadas no PMLR. A ordem na qual as variáveis são fixadas apresenta grande importância para a solução, uma vez que pode resultar em uma solução de boa qualidade, ou até mesmo uma solução infactível. As primeiras variáveis a serem fixadas são w_n^b , pois elas restringem o espaço de soluções das demais variáveis sem originar uma solução infactível. Com as variáveis w_n^b fixadas, a complexidade da alocação do berço (v_{nt}^b) é reduzida e as possibilidades de alocação dos produtos nas subáreas (f_{pt}^s) é limitada. As variáveis w_n^b também apresentam a particularidade de ter um número menor de variáveis, pois estão associadas a apenas dois conjuntos. Com essa característica é possível realizar uma busca mais ampla no espaço de soluções, sendo exatamente essa vantagem que será explorada na heurística aplicada nestas variáveis. Tanto as variáveis w_n^b , como as v_{nt}^b e f_{pt}^s não fazem parte do subproblema de pricing, portanto, fixá-las em 0 ou 1 não afeta a estrutura do subproblema. Consequentemente, não é necessária nenhuma alteração na formulação do subproblema durante a resolução dos nós da árvore.

Com objetivo de encontrar uma solução inteira viável para as variáveis w_n^b , foi desenvolvida uma heurística de mergulho com *backtracking* (DHB-I). Essa heurística é

baseada nos trabalhos de Harvey e Ginsberg (1995) e Sadykov et al. (2019). A estratégia consiste em realizar um número limitado de busca em profundidade e utilizar a melhor solução encontrada.

A heurística DHB-I é dividida em duas fases: na fase 1, considere como nó inicial a solução ótima do PMLR (após a convergência do procedimento de geração de colunas). A partir do nó inicial é realizada uma busca em profundidade, com a intenção de encontrar uma solução inteira viável (*depth-first search*). A profundidade das buscas são limitadas pelo parâmetro *maxdepth*, que é definido previamente. Em cada nó da árvore, uma ramificação é formada com base na estratégia de arredondamento, que fixa em 1 as variáveis fracionárias que tiverem um valor maior que o parâmetro Ω . Caso nenhuma variável seja fixada, o valor de Ω é reduzido (em intervalos de tamanho constante igual ao valor do parâmetro ϵ), até que pelo menos uma variável seja fixada. Em seguida, inicia novamente o processo de geração de colunas: resolve a cada iteração o PMLR e o subproblema de precificação até não haver mais colunas com custo reduzido menor que zero para serem inseridas. Este processo de fixar uma parte das variáveis e procedimento de geração de colunas é repetido em todos os nós da árvore, até que seja encontrada uma solução inteira. Caso chegue no último nó e a solução encontrada ainda não seja inteira, as últimas variáveis fracionárias que tiverem o valor mais próximo de 1 são fixadas, e, em seguida, realizado o processo de geração de colunas para gerar as últimas colunas. A Figura 8 apresenta um fluxograma do algoritmo da primeira fase da heurística DHB-I.

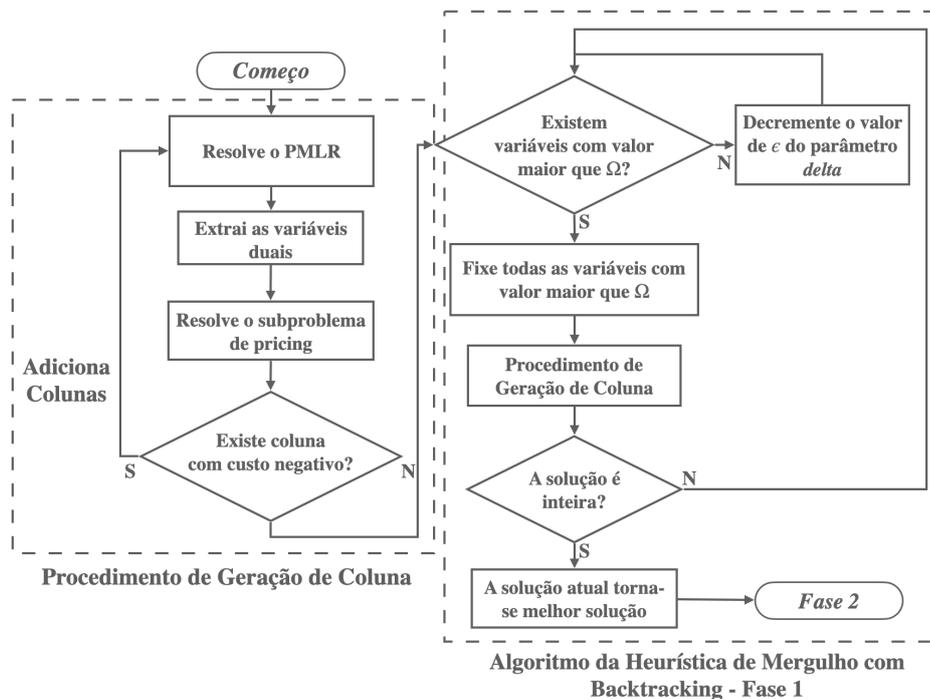


Figura 8 – Fluxograma do procedimento de geração de colunas e da fase 1 da heurística de mergulho com backtracking.

Na fase 2 da heurística DHB-I são realizados os *backtracking* em cada nó formado na primeira fase. Novas ramificações são geradas em cada um dos nós gerados anteriormente. Nessas novas ramificações, as variáveis que foram fixadas em 1 são então fixadas em 0, as demais variáveis que estavam relaxadas e as que foram fixadas em 1 em nós anteriores, permanecem do mesmo modo. A Figura 9 ilustra parte de uma árvore gerada pela heurística DHB-I e exemplifica o funcionamento dos *backtrackings*, no qual o conjunto A representa as variáveis fixadas em 1 no primeiro nó na primeira fase, e o conjunto B as variáveis fixadas na ramificação seguinte na mesma fase. Ainda nesta figura, os nós destacados em cinza escuro representam a fase 1 (*depth-first search*), e os nós em cinza claro representam a fase 2 da heurística (*backtrackings*).

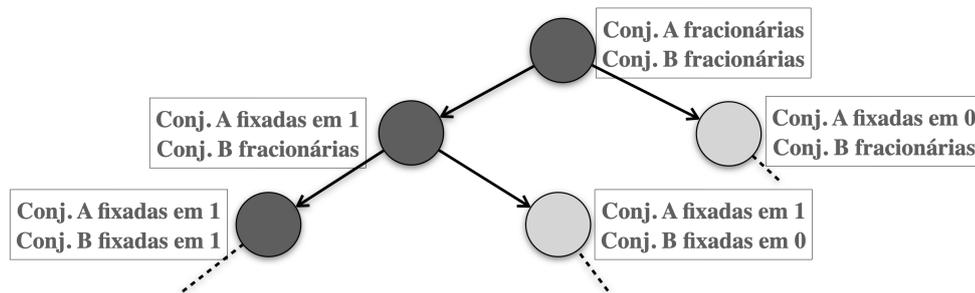
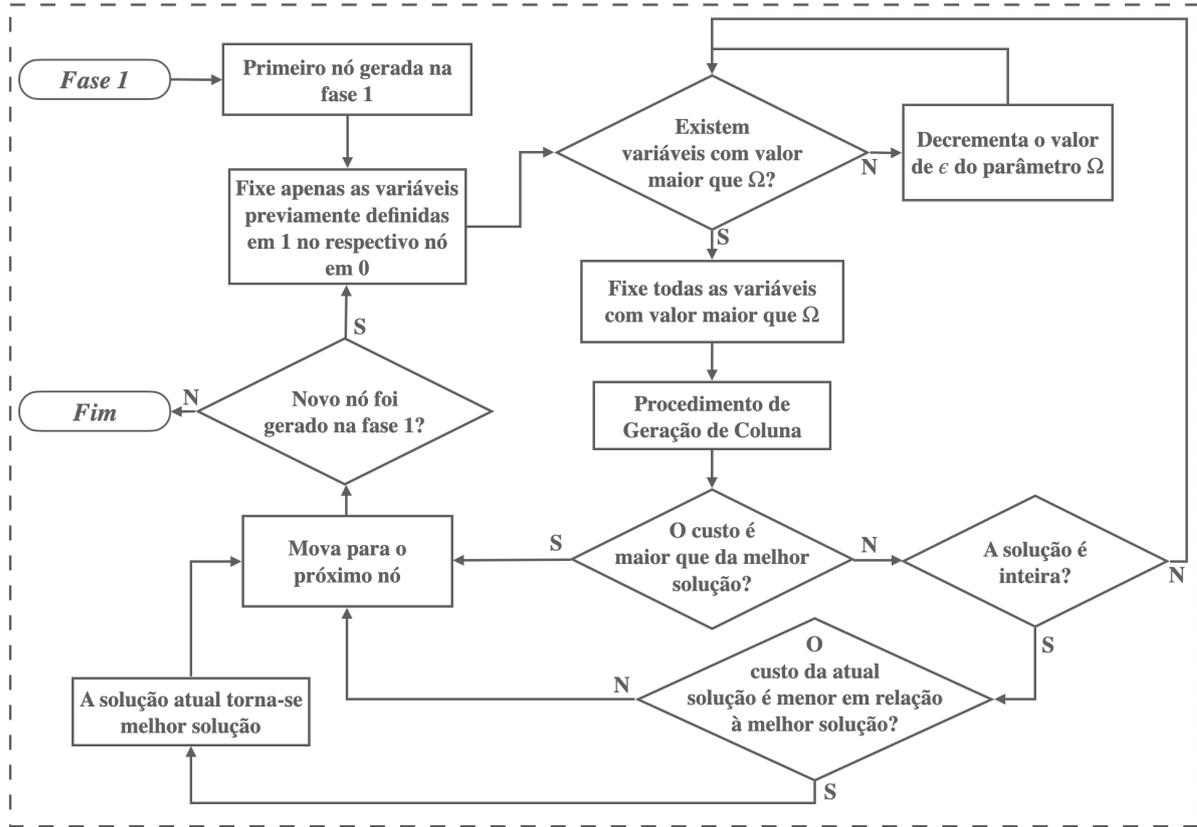


Figura 9 – Exemplo de árvore de enumeração da heurística de mergulho com *backtracking*.

Depois de fixadas as específicas variáveis em zero, repete-se os mesmos procedimentos fixação de variáveis e geração de colunas da fase 1, pretendendo obter uma solução inteira factível para as variáveis w_n^b . Esse procedimento de *backtracking* é repetido para todos os nós gerados na primeira fase da heurística, obtendo então no final da fase 2 várias soluções inteiras, em que a solução com o menor custo é selecionada. A Figura 10 apresenta um fluxograma do algoritmo da segunda fase da heurística DHB-I.

A solução obtida no final da primeira fase é armazenada como melhor solução. Caso uma nova solução inteira viável seja obtida na segunda fase e possua um custo (valor da função objetivo) inferior, a melhor solução é alterada. Para melhorar o desempenho da heurística, caso um custo obtido durante os *backtrackings* seja superior em relação a melhor solução, a busca em profundidade é encerrada para determinado nó. No entanto, a heurística continua funcionando, realizando *backtracking* no nó seguinte.

O parâmetro Ω pode ter um valor entre 0 e 1, em que quanto mais próximo do valor 1 menos variáveis serão fixadas em cada nó, logo as buscas em profundidade apresentarão mais nós, e conseqüentemente mais *backtracking* serão realizado, possibilitando um maior número de ramificações e uma busca mais larga. Em contrapartida, o tempo computacional deve ser maior do que um Ω com valor inferior, pois a árvore de enumeração será menor. Portanto, o valor do parâmetro Ω influencia diretamente na busca.



Algoritmo da Heurística de Mergulho com Backtracking - Fase 2

Figura 10 – Fluxograma da fase 2 da heurística de mergulho com *backtracking*.

4.2.3.2 Heurística *Relax-and-Fix* de Arredondamento

Após as variáveis w_n^b serem fixadas de acordo com a melhor solução inteira gerada pela heurística DHB-I, as variáveis de alocação de subárea (f_{pt}^s), que representam qual produto p será armazenado na subárea s no período t , serão fixadas. Para obter uma solução inteira para as variáveis f_{pt}^s , foi desenvolvido novamente um procedimento heurístico de arredondamento, nomeado neste trabalho como heurística *relax-and-fix* de arredondamento (RRFH). A heurística RRFH desenvolvida é baseada na heurística *relax-and-fix* proposta no trabalho de Wolsey (1998). A lógica do algoritmo desta heurística é baseada na resolução de uma sequência de subproblemas que mantém um conjunto de variáveis fixadas, um conjunto relaxado e apenas um conjunto integralizado.

Na solução parcialmente relaxada, as variáveis f_{pt}^s estão fracionárias em todos os períodos do horizonte de planejamento. O algoritmo da heurística RRFH trabalha fixando variáveis com valor igual ou mais próximo de 1 de um determinado período em cada iteração. Após as variáveis serem fixadas, inicia-se novamente o processo de geração de colunas. Em um procedimento iterativo, repetido para cada período, as subáreas são alocadas de apenas um produto em todo horizonte. Ao final da heurística RRFH é obtida uma solução inteira

para as variáveis f_{pt}^s . O pseudocódigo da heurística RRFH é apresentado no Algoritmo 1.

Algoritmo 1: *Heurística relax-and-fix de arredondamento*

Result: Fixar todas as variáveis f_{pt}^s .

```

1 solução ← heuristicaMergulhoBacktracking()
2 for t = 1 → T do
3   for s = 1 → S do
4     δ → 0
5     for p = 1 → P do
6       if  $f_{pt}^s > \delta$  then
7         δ ←  $f_{pt}^s$ 
8         fp ← p
9       end
10    end
11     $f_{(fp)t}^s = 1$ 
12  end
13  solução ← procedimentoGeracaoColunas()
14 end

```

Com esta estratégia de fixação por período, as soluções dos períodos seguintes podem ser alteradas a cada iteração, permitindo então alocar produtos em subáreas que antes seriam atribuídas a outros produtos. Esse importante detalhe pode proporcionar uma solução de melhor qualidade, que seria deixada de lado caso adotasse uma estratégia mais simples, como exemplo fixar todas as variáveis próximas de 1 inicialmente.

As variáveis w_n^b previamente fixadas atribuíram os berços para cada navio, no entanto as variáveis v_{nt}^b , que definem a sequência e o tempo de atendimento dos navios nos berços, ainda permanecem fracionárias. O procedimento de fixação destas variáveis requer um certo cuidado, visto que os 1's fixados devem ser consecutivos, caso contrário as restrições (7) (que impedem a preempção no carregamento dos navios) serão infringidas, tendo assim uma solução infactível. Com a intenção de viabilizar uma solução inteira factível, foi utilizado o algoritmo *branch-and-cut* do solver CPLEX. A estratégia consiste em converter as variáveis v_{nt}^b para binário, aplicar o algoritmo *branch-and-cut*, fixar a solução obtida, e realizar o procedimento de geração de colunas para obter as últimas colunas. A Figura 11 apresenta um fluxograma do procedimento. O uso do algoritmo do *solver* é somente possível pois durante o processo de fixação das variáveis w_n^b e f_{pt}^s , foram geradas inúmeras colunas e, intencionalmente, nenhuma coluna foi removida.

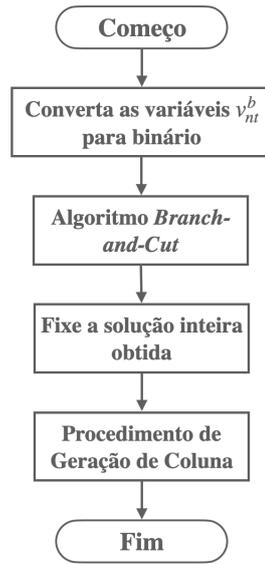


Figura 11 – Fluxograma do procedimento de fixação das variáveis v_{nt}^b .

Capítulo 5

Abordagem de solução para o PIPSAPAB-II

O principal objetivo deste capítulo é apresentar uma estratégia de solução para a formulação matemática PIPSAPAB-II que seja capaz de produzir um bom limite superior.

5.1 Introdução

Do mesmo modo que a formulação PIPSAPAB-I, solucionar a formulação PIPSAPAB-II com instâncias de larga escala via *solver*, como CPLEX, é uma tarefa extremamente difícil. Para contornar estas limitações e obter boas soluções, foi desenvolvido um algoritmo que combina algumas heurísticas matemáticas (CHM).

O algoritmo consiste basicamente em encontrar uma boa solução para um conjunto de variáveis binárias, fixar essa solução, repetir esse procedimento com outros conjuntos específicos de variáveis até obter uma solução inteira factível. Primeiro são fixadas as variáveis w_n^b (responsáveis por alocar um berço para cada navio) através da heurística de mergulho com *backtracking*. Após fixadas as variáveis w_n^b , são aplicadas duas heurísticas (*relax-and-fix* e *local branching*) para obter e fixar uma solução inteira para as variáveis v_{nt}^b (responsáveis por atribuir um ou mais períodos para o carregamento dos navios nos berços). Com as variáveis w_n^b e v_{nt}^b fixadas, são executadas duas novas heurísticas (*relax-and-fix* e *local branching*) para somente fornecer uma solução inteira factível para as variáveis f_{pt}^s (responsáveis por alocar os produtos nas subáreas), que é utilizada na última heurística do algoritmo. Por último, é implementada uma heurística *rolling horizon* para fixar as variáveis f_{pt}^s e q_{pit}^r (responsáveis por sequenciar as rotas e impedir que as mesmas compartilhem equipamento), e assim obter uma solução inteira para a formulação PIPSAPAB-II. A Figura 12 exibe um fluxograma para estratégia adotada.

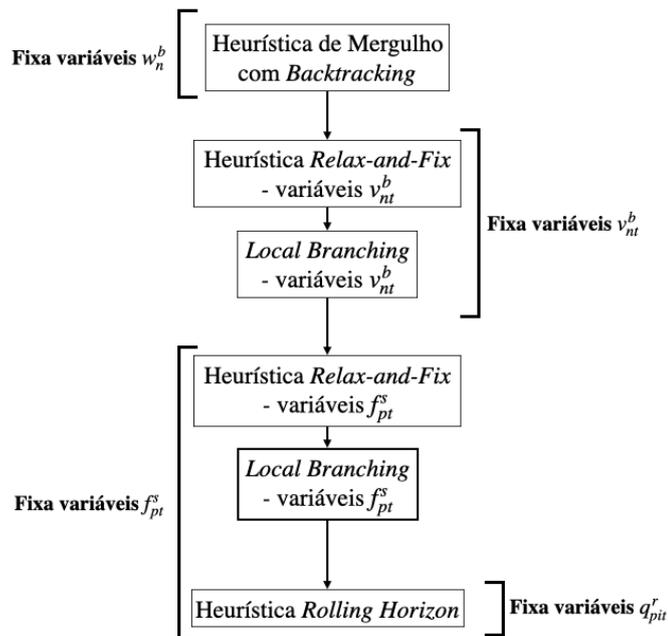
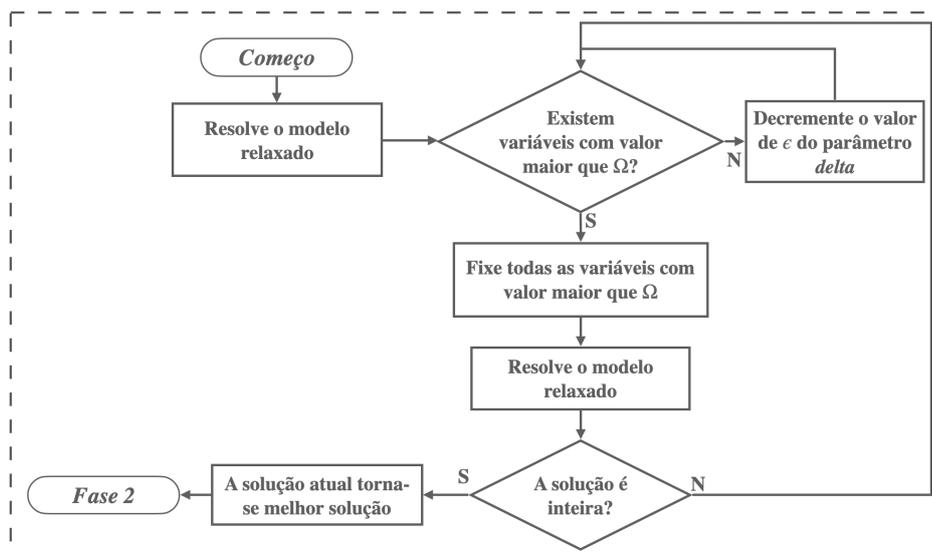


Figura 12 – Fluxograma do algoritmo CMH.

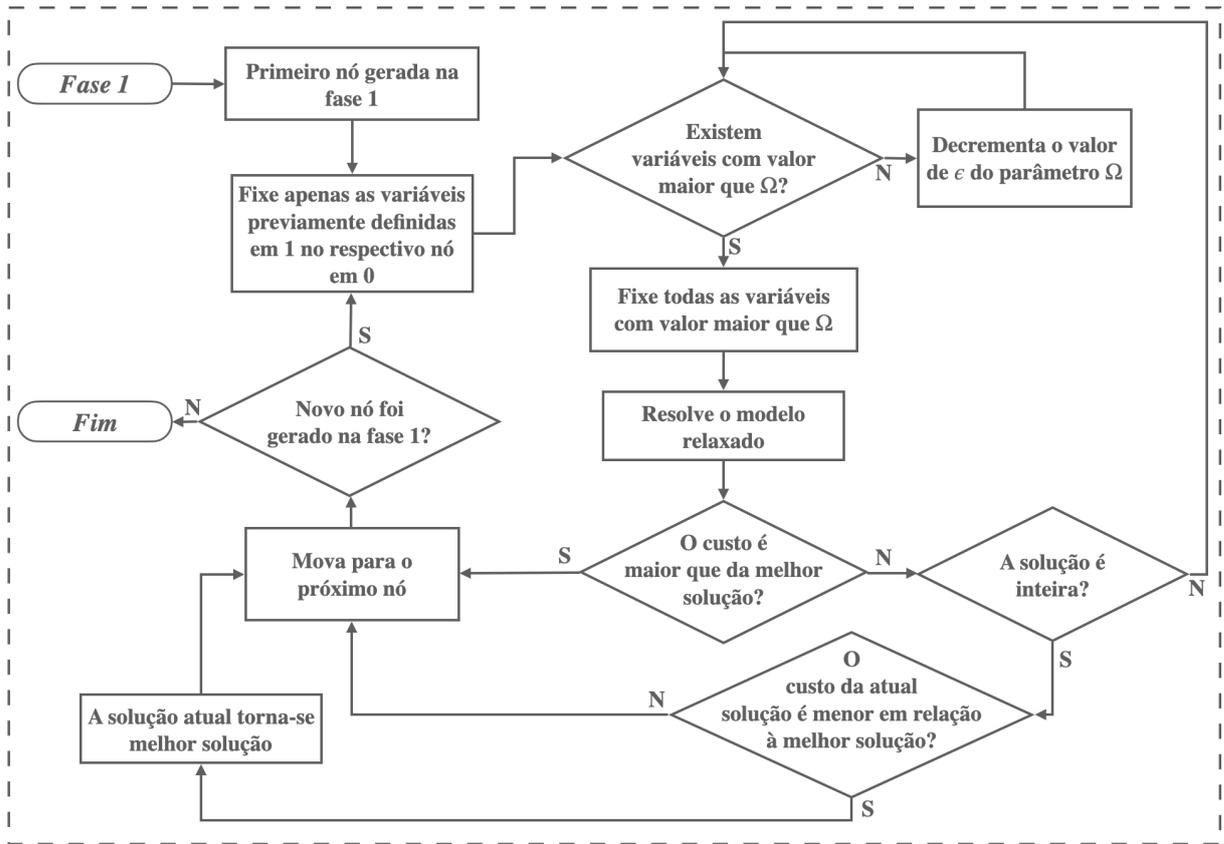
5.2 Heurística de Mergulho com *Backtracking*

Como apresentado no fluxograma da Figura 12, as primeiras variáveis a serem fixadas são as w_n^b . Essas variáveis terão os valores fixados de acordo com a solução obtida por uma heurística de mergulho com *backtracking*. As Figuras 13 e 14 apresentam o fluxograma desta heurística.



Algoritmo da Heurística de Mergulho com *Backtracking* - Fase 1

Figura 13 – Fluxograma da fase 1 da heurística de mergulho com *backtracking*.



Algoritmo da Heurística de Mergulho com Backtracking - Fase 2

Figura 14 – Fluxograma da fase 2 da heurística de mergulho com *backtracking*.

A heurística de mergulho com *backtracking* implementada pelo algoritmo CHM é semelhante a DHB-I, uma vez que ambas apresentam a mesma estratégia. Além disso, a comparação dos fluxogramas das Figuras 13 e 14, em relação às Figuras 8 e 10, demonstram a proximidade entre as duas heurísticas. Porém, elas apresentam certas diferenças. Para diferenciar as duas heurísticas, a heurística apresentada nesta seção será identificada como DHB-II. A diferença entre a DHB-I e DHB-II está vinculada à geração de colunas. Diferente da formulação PIPSAPAB-I, a formulação PIPSAPAB-II não é decomposta em problema mestre e subproblema, logo não utiliza o método de geração de colunas. A heurística DHB-I se inicia com o procedimento de geração de colunas de microperíodos para PMLR (Figura 8), enquanto que o primeiro passo da fase 1 da DHB-II é resolver o modelo compacto PIPSAPAB-II com todas as variáveis binárias relaxadas, como mostra a Figura 13. Além dessa diferença, a DHB-I repete o procedimento de geração de colunas inúmeras vezes na primeira e segunda fase, enquanto que a DHB-II soluciona o modelo relaxado, ao invés de realizar esse procedimento. A Figura 14 demonstra que, após fixadas as variáveis com valor superior a Ω , o modelo relaxado é solucionado via CPLEX.

Mesmo sem o procedimento de geração de colunas, a heurística DHB mantém-se

como um vantajoso algoritmo para obter soluções para as variáveis w_n^b , pois não foram feitas alterações nas restrições no qual essas variáveis fazem parte, logo as vantagens destacadas anteriormente (início da seção 4.2.3.1) permanecem.

5.3 Heurística *Relax-and-Fix* para as variáveis v

Após fixadas as variáveis w_n^b com a melhor solução encontrada pela heurística de mergulho com *backtracking*, são executadas uma heurística *relax-and-fix* para obter uma solução inteira inicial factível, e, em seguida, heurística *local branching* para refinar a solução das variáveis v_{nt}^b . O Algoritmo 2 apresenta o pseudocódigo da uma heurística *relax-and-fix* específica para as variáveis v_{nt}^b (HRF-V).

A estratégia da heurística HRF-V consiste basicamente em fixar as variáveis por período, seguindo o critério de fixar em 1 as variáveis com o valor mais próximo de 1, em seguida verificar a quantidade carregada pelo navio e fixar o período seguinte, caso a porcentagem mínima de carregamento não seja atingida.

Como entrada para a heurística HRF-V, é adicionado o vetor vec_{max} , que tem a responsabilidade de limitar o número de períodos que podem ser atribuídos para cada um dos navios. Esse vetor é utilizado como estratégia para contornar a geração de soluções infactíveis durante o procedimento de fixação das variáveis v_{nt}^b . O vetor possui um valor para cada navio disponível no horizonte de planeamento. Esses valores são baseados na média de períodos em cada navio para fazer carregamentos. Esse valor médio é dado pela seguinte expressão:

$$\bar{t} = \left\lfloor \frac{|B| \times |T|}{|N|} \right\rfloor$$

Para os navios maiores é dado $\bar{t} + 1$ períodos, enquanto que, para os navios médios, é dado \bar{t} ; e, para os navios menores, é dado $\bar{t} - 1$. Note que esta heurística é gulosa.

Esta heurística é dividida em iterações, sendo que o número de iterações é igual ao de períodos. Em cada iteração existem dois procedimentos de fixação de variáveis. Primeiramente, o modelo relaxado é solucionado pelo *solver*; em seguida inicia-se a primeira iteração. No início dessa iteração ocorre a primeira etapa de fixação, que consiste em fixar em 1 as variáveis que tiverem o maior valor em cada berço no período 1, desse modo alocando apenas um navio em cada berço no período da iteração. O procedimento descrito está entre as linhas 4 e 14 do Algoritmo 2. Depois de fixadas, o modelo é novamente solucionado pelo *solver*, e assim inicia-se a segunda etapa de fixação. Nesta etapa, cada navio que foi fixado na etapa 1 tem a quantidade de produto carregado até o período da iteração verificada. Caso a quantidade carregada pelo determinado navio for menor que 70% da carga total e o número de períodos atribuídos para o respectivo navio for

inferior ao máximo (dado pelo vec_{max}), é fixado em 1 a variável v_{nt}^b associada ao período 2. O procedimento descrito está entre as linhas 16 a 24 do Algoritmo 2. Por exemplo, se o navio 2 possuir o maior valor entre os navios alocados no berço 3 no período 1, a variável v_{21}^3 é fixada em 1; após o modelo ser solucionado, é verificada a quantidade de produto carregado no período 1, caso for inferior a 70% e a quantidade de períodos dado pelo vetor vec_{max} for igual ou superior a 2, a variável v_{22}^3 é fixada em 1, caso contrário não é fixada. Depois de finalizada a segunda etapa de fixação da iteração 1, inicia-se a segunda iteração.

Na segunda iteração e nas seguintes iterações, são repetidos os mesmos procedimentos da iteração 1, ou seja, a primeira etapa de fixação, solução do modelo relaxado via *solver* e segunda etapa de fixação. Com a diferença que serão fixadas em 1 na primeira etapa do período t (ou iteração t) apenas as variáveis dos berços que não tiveram navios fixados na segunda etapa da iteração anterior (período $t - 1$). Além disso, todas as variáveis fixadas em 1, sejam elas fixadas na primeira etapa do período atual t ou fixadas na segunda etapa do período $t - 1$, serão verificadas e fixadas na segunda etapa do período t . No exemplo anterior, as variáveis v_{21}^3 e v_{22}^3 do navio 2 alocado no berço 3, foram fixadas na iteração; já na segunda iteração, nenhuma variável é analisada na primeira etapa, mas, na segunda etapa são verificadas as variáveis desse navio e berço. Note que a segunda etapa de fixação não é realizada no último período.

Algoritmo 2: *Heurística relax-and-fix para variáveis v_{nt}^b*

Entradas: solução inicial s com as variáveis w_n^b fixadas, vetor $vec_{max}[n]$ com o número máximo de período atribuídos para cada navio n , vetor $vec[n]$ contador de períodos fixados por cada navio.

```
1  $vec[n] \leftarrow [0, \dots, 0]$  ▷ Vetor contador inicialmente vazio.
2  $s \leftarrow resolveModeloRelaxado();$ 
3 for  $t = 1 \rightarrow T$  do
4   for  $b = 1 \rightarrow B$  do
5      $\epsilon \leftarrow 0$ 
6     for  $n = 1 \rightarrow N$  do
7       ▷ Loop for para verificar o navio  $n$  que possui o maior valor.
8       if  $\epsilon < v_{nt}^b$  then
9          $\epsilon = v$ 
10         $n_b = n$ 
11      end
12    end
13     $v_{n_b t}^b = 1$  ▷ Fixa em um a variável  $v$ , associado ao navio  $n$ , que tiver o maior valor no berço  $b$  em 1.
14     $vec[n_b] = vec[n_b] + 1$  ▷ Soma mais um no navio fixado no vetor contador.
15  end
16   $s \leftarrow resolveModeloRelaxado();$ 
17  if  $t \leq (T - 1)$  then
18    for  $b = 1 \rightarrow B$  do
19      verifique a quantidade de produto carregado pelo o navio  $n_b$  correspondente a variável  $v_{nt}^b$  do berço  $b$  fixada anteriormente até o período  $t$ ;
20      if quantidade produto carregado por  $n_b$  é menor que 70% da carga total do navio and  $vec_{n_b} < vec_{max}[n_b]$  then
21         $v_{n_b(t+1)}^b = 1$ 
22         $vec[n] = vec[n] + 1$ 
23      end
24    end
25 end
```

5.4 Local Branching

A partir de uma solução inicial dada pela heurística *relax-and-fix* para as variáveis v_{nt}^b , é implementada a heurística matemática *local branching*, com o objetivo de melhorar a solução. A *matheurística local branching* é uma técnica introduzida por Fischetti e Lodi (2003), com o objetivo de abordar problemas inteiros de larga escala. Na literatura

é possível encontrar diversos trabalhos que demonstram a eficiência desta técnica para solucionar problemas reconhecidamente difíceis, como: *open pit mine production scheduling problem* em [Samavati et al. \(2017\)](#); problema de roteamento de veículos com demanda em [Hernandez et al. \(2019\)](#); *graph edit distance problem* em [Darwiche et al. \(2019\)](#); problema de sequenciamento de máquinas paralelas idênticas [Schmidt et al. \(2019\)](#); problema de localização de facilidade em [Yang et al. \(2019\)](#). A *matheurística local branching* combina uma abordagem exata com ideias de busca em vizinhanças utilizando *solvers* comerciais de modelo de programação inteira mista. O conceito de busca local está associado com a introdução de cortes, ou inequações, na formulação matemática original. Inicialmente, o objetivo consiste em explorar regiões menores com um menor esforço computacional do que o problema original, e mesmo assim obter uma solução de boa qualidade. No algoritmo geral do *local branching* proposto em [Fischetti e Lodi \(2003\)](#), o problema (P) pode ser descrito a seguir.

$$\begin{aligned}
(P) \quad & \mathbf{min} \quad c^T x \\
& \mathbf{sujeito \ a} : \quad Ax = b \\
& x_j \in \{0, 1\} \quad \forall j \in \beta \neq 0 \\
& x_j \geq 0 \quad \forall j \in \delta
\end{aligned}$$

Neste problema, o conjunto de variáveis x_j é dividido em β e δ , sendo β o conjunto de variáveis binárias e δ as variáveis relaxadas. Dada uma solução inteira factível \bar{x} para o problema (P) e o parâmetro inteiro k , a k -vizinhança $\Delta(\bar{x}, x)$ de \bar{x} representa o conjunto de soluções factíveis de (P) que satisfazem a restrição de *local branching*.

$$\Delta(\bar{x}, x) = \sum_{j \in \beta: \bar{x}_j=1} (1 - x_j) + \sum_{j \in \beta: \bar{x}_j=0} x_j \leq k$$

Esta restrição pode ser incorporada aos algoritmos *Branch-and-Cut* como estratégia de ramificação para explorar o espaço de soluções de (P) . Dada a solução incumbente \bar{x} , o espaço de soluções pode ser particionado em dois conjuntos disjuntos com adição das restrições $\Delta(\bar{x}, x) \leq k$ (ramificação à esquerda) e $\Delta(\bar{x}, x) \geq k + 1$ (ramificação à direita).

A ação da *matheurística local branching* inicia-se explorando o subproblema originado com a adição das restrições de ramificação à esquerda. O parâmetro k define o tamanho da vizinhança, logo ele deve ser suficientemente pequeno para que o subproblema originado seja solucionado em um tempo relativamente curto; no entanto, deve ser grande o suficiente para gerar soluções melhores do que \bar{x} . Além disso, espera-se que o subproblema correspondente com a ramificação à esquerda seja mais fácil de resolver do que o problema anterior sem a restrição. O trabalho de [Fischetti e Lodi \(2003\)](#) apresenta o algoritmo clássico de *local branching*. O algoritmo *local branching* desenvolvido neste trabalho é apresentado no Algoritmo 3.

Algoritmo 3: Local Branching

Entradas: solução inicial viável s' , distância mínima k_0 , tempo limite de execução do solver t_{limit} .

```
1  $s^* \leftarrow s'$ 
2  $k \leftarrow k_0$ 
3  $it \leftarrow 0$ 
4 while  $it \leq it_{max}$  do
5   | adicione a restrição  $\Delta(s^*, s') \leq k$  ao modelo;
6   |  $s' \leftarrow resolveModelo(t_{limit})$ 
7   | if custo da solução viável  $s'$  é menor do que solução viável  $s^*$  then
8   |   | remova a restrição  $\Delta(s^*, s') \leq k$  do modelo;
9   |   | adicione a restrição  $\Delta(s^*, s') \geq k + 1$  ao modelo;
10  |   |  $s^* \leftarrow s'$ 
11  | end
12  | else
13  |   | remova a restrição  $\Delta(s^*, s') \leq k$  do problema P;
14  |   |  $k = k + 1$ 
15  | end
16  |  $it = it + 1$ 
17 end
```

Inicialmente, a solução s' obtida anteriormente na heurística *relax-and-fix* é utilizada como uma solução de referência (s^*). Então adiciona-se uma restrição de *local branching* da forma $\Delta(s^*, s') \leq k$ ao modelo, e, em seguida, o solver *CPLEX* é executado até encontrar uma solução ótima ou até atingir o tempo limite de execução (t_{limit}). Caso esta nova solução (s') tenha um custo menor do que a solução s^* , a última restrição de *local branching* da forma $\Delta(s^*, s') \leq k$ é removida, uma restrição de *local branching* da forma $\Delta(s^*, s') \geq k + 1$ é adicionada na formulação, significando que será explorado uma ramificação do lado direito da árvore de enumeração do *local branching*, e retorna o valor de k para a distância mínima. Caso a solução s' não tenha um custo menor que a solução s^* , a última restrição de *local branching* da forma $\Delta(s^*, s') \leq k$ é removida e o valor do parâmetro k é incrementado em 1, e assim ampliando a distância. Posteriormente a qualquer um dos dois casos, uma nova restrição de *local branching* da forma $\Delta(s^*, s') \leq k$ baseada na nova solução encontrada, ou no novo valor de k . Esse procedimento é repetido até atingir a iteração máxima. Ambas restrições de *local branching* permanece igual em relação a proposta em (FISCHETTI; LODI, 2003).

5.5 Heurística *Relax-and-Fix* para as variáveis f_{pt}^s

Com as soluções inteiras das variáveis w_n^b e v_{nt}^b já definidas, são iniciados os procedimentos para obtenção de uma solução inteira viável para as variáveis f . Primeiramente será executada uma heurística *relax-and-fix* específica para as variáveis f_{pt}^s com objetivo de obter uma solução inicial, depois o mesmo algoritmo de *local branching* das variáveis v_{nt}^b , porém com parâmetros diferentes, e por último, a heurística *rolling horizon* para determinar uma solução integrada das variáveis f_{pt}^s e q_{pit}^r .

Nesta seção, é apresentada e explicada a heurística *relax-and-fix* específica para as variáveis f_{pt}^s (HRF-F). O pseudocódigo para esta heurística é apresentado no Algoritmo 4.

Algoritmo 4: Heurística *relax-fix* para variáveis f_{pt}^s

Entradas: solução inicial s com as variáveis w_n^b e v_{nt}^b fixadas.

```

1  $\alpha \leftarrow 1$ 
2 while  $\alpha > 0.5$  or  $s \notin \mathbb{Z}$  do
3    $s \leftarrow \text{resolvaModeloRelaxado}();$ 
4   for  $t = 1 \rightarrow T$  do
5     for  $s = 1 \rightarrow S$  do
6       for  $p = 1 \rightarrow P$  do
7         if  $f_{pt}^s \geq \alpha$  then
8            $f_{pt}^s = 1$            ▷ Se a variável  $f_{pt}^s$  for maior que  $\alpha$  é fixada em 1.
9         end
10      end
11    end
12  end
13   $\alpha = \alpha - 0.05$ 
14 end
15 if  $s \notin \mathbb{Z}$  then
16   converte as variáveis  $f_{pt}^s$  ainda não fixadas em binário;
17    $s \leftarrow \text{resolvaModeloRelaxado}();$ 
18   fixe a solução obtida para as variáveis  $f_{pt}^s$ ;
19 end

```

Esta heurística *relax-and-fix* é dividida em duas partes. A primeira parte inicia-se solucionando o modelo com as variáveis f_{pt}^s relaxadas (como também as variáveis q_{pit}^r) por meio do *solver* CPLEX. A partir da solução obtida, as variáveis f_{pt}^s que tiverem um valor superior ao parâmetro α são fixadas em 1, desse modo alocando um produto p na subárea s em um período t . Em seguida, o parâmetro α é decrementado pelo valor 0,05. Esses três procedimentos são repetidos em um *loop* até que uma solução inteira factível seja obtida,

ou o valor de α seja inferior a 0,5. Esse valor limitante é justificado pelo possível caso de ocorrer duas ou mais variáveis f_{pt}^s terem valor superior de α em apenas uma subárea. Se essas variáveis fossem fixadas, originaria uma solução infactível, pois as restrições (14) não seriam respeitadas e então permitindo alocar mais de um produto em apenas uma subárea. O procedimento descrito está entre as linhas 2 e 14 do Algoritmo 4.

Caso não for encontrada uma solução inteira para as variáveis de alocação de produto nas subáreas, inicia-se a segunda parte da heurística, que consiste em converter para binário as variáveis f_{pt}^s que ainda não foram fixadas, resolver o modelo através do *solver* e fixar a solução gerada. O procedimento descrito está entre as linhas 15 e 19 do Algoritmo 4.

A solução inteira factível encontrada pela heurística *relax-and-fix* do algoritmo 6 será utilizada como solução incumbente para um novo procedimento de *local branching* para apenas as variáveis f_{pt}^s .

5.6 Heurística *Rolling Horizon*

A solução inteira para as variáveis f_{pt}^s obtida pela aplicação do *local branching* é fixada, mas será utilizada apenas como base para solução final destas variáveis. Pois a solução inteira para as variáveis f_{pt}^s e q_{pit}^r será obtida de forma integrada utilizando uma heurística *rolling horizon* (HRH).

A heurística *rolling horizon* é amplamente em problemas de dimensionamento de lotes e planeamento de produção, mas também tem sido utilizada em *multi-period network design and routing problem* em Papadimitriou e Fortz (2015), problema integrado de sequenciamento de guindastes e alocação de berço em Agra e Oliveira (2017), *cargo mix problem* em Christensen, Erera e Pacino (2019). A principal ideia desta heurística é particionar o horizonte de planeamento em subproblemas menores, e em seguida, resolver consecutivamente cada um dos subproblemas. Com essa estratégia pretende-se reduzir o esforço computacional, pois apenas um subproblema apresentará variáveis binárias, enquanto que os subproblemas que foram solucionados apresentam variáveis fixadas e os subproblemas seguintes as variáveis estão relaxadas.

Neste trabalho, cada subproblema será um período t . A estratégia da heurística *rolling horizon* consiste em repetir o seguinte procedimento para cada período t : converter somente as variáveis f e q em binária do período t , solucionar o modelo através *solver* de maneira ótima, e fixar a solução obtida. Mas note que, enquanto as variáveis f_{pts} e q_{pit}^r são binárias do período t , no período $t + 1$ as variáveis f_{pts} estão fixadas seguindo a solução obtida pelo *local branching* e as variáveis q_{pit}^r estão relaxadas. Além disso, as variáveis f_{pts} e q_{pit}^r no período $t - 1$ estão fixadas segundo a solução da HRH. O pseudocódigo para esta

heurística é apresentado no Algoritmo 5.

Algoritmo 5: *Heurística Rolling Horizon*

Entradas: solução inicial s com as variáveis w_n^b , v_{nt}^b e f_{pt}^s fixadas.

1 for $t = 1 \rightarrow T$ **do**

2 remova as fixações das variáveis f_{pt}^s no período t ;

3 converta as variáveis f_{pt}^s e q_{pit}^r do período t para binário;

4 resolva o modelo através do *solver CPLEX*;

5 fixe a solução obtida para as variáveis f_{pt}^s e q_{pit}^r do período t ;

6 end

Com a conclusão desta última heurística é obtido uma solução inteira factível para o PIPSAPAB-II.

Capítulo 6

Resultados Computacionais

Neste capítulo serão discutidos os experimentos computacionais realizados utilizando os modelos matemáticos e as abordagens de solução apresentados anteriormente.

6.1 Gerador de Instâncias

Para avaliar o desempenho dos algoritmos propostos, foi desenvolvido um gerador de instâncias para simular diversas características de terminal portuário de carga a granel, tais como: número de berços de atracação, quantidade e configuração das rotas, horizonte de planejamento, número e capacidade de equipamentos, oferta, demanda dos navios, entre outros. Os objetivos são gerar instâncias que abrangem um conjunto de configurações que sejam próximas da realidade e que estressem ao máximo os métodos de solução desenvolvidos. Uma parte dos parâmetros utilizados é detalhado na Tabela 6.

Tabela 6 – Dados utilizados para gerar todas as instâncias.

Parâmetro	Descrição
Pátios	A área de armazenamento de produtos contém 5 pátios. Cada pátio foi dividido em 10 subáreas, e cada uma tem a capacidade limite de 100.000 toneladas de armazenamento. No total, as 50 subáreas podem armazenar 5.000.000 toneladas de produto.
Equipamentos	Cinco viradores de vagão, cinco empilhadeiras de minério, quatro recuperadores /empilhadeiras, cinco recuperadores e seis carregadores de navios. Além desses, são utilizados aproximadamente 50 km de correias transportadoras. A capacidades dos equipamentos variam entre 12.000 e 16.000 toneladas/hora.
α_{pt}	0.5 (unidades monetárias) para todos os produtos e períodos.
β_{np}	1 (uma unidade monetária) para todos os navios e produtos.
ϕ_n	10 (dez unidades monetárias) para todos os navios.
$\gamma_{pp't}^s$	1 (uma unidade monetária) para qualquer produto trocado em todos os períodos e subáreas.
σ^r	Baseado na seguinte formula: 0.01 (unidades monetárias) \times comprimento da rota r .

A prioridade do terminal é atender a demanda. Logo o parâmetro associado ao não atendimento da demanda (β_{np}) deve ser maior que o associado ao não atendimento

da oferta (α_{pt}), para gerar maior penalidade. Baseado nisto, assumimos que o valor de β_{np} deve ser o dobro de α_{pt} . Já o parâmetro ϕ_n apresenta o maior valor em relação aos termos da função objetivo, pois um dos principais objetivos é reduzir o tempo de serviço. E finalmente, o parâmetro σ_r é calculado com base no comprimento da rota. Com relação aos valores de oferta que devem ser atendidos em cada período, estes valores foram gerados aleatoriamente, mas baseados em dados reais de um terminal portuário de granéis sólidos. A quantidade de pátios, equipamentos e as demais características buscam retratar condições próximas da realidade de um terminal portuário graneleiro. Os demais parâmetros utilizados para gerar as instâncias serão descritos posteriormente e de acordo com a formulação.

Todos os experimentos foram conduzidos em um computador com processador Intel XEON E51620v2 CPU 3.50GHz e memória física de 32 GB, executando a versão 12.9 do solver CPLEX. Para todas as instâncias, foi estabelecido um limite de tempo de 6 horas (21600 segundos).

6.2 Resultados para CGBH na solução do PIPSAPAB-I

Para os experimentos computacionais da CGBH na solução da formulação PIPSAPAB-I, além dos apresentados anteriormente, são também considerados os parâmetros mostrados na Tabela 7.

Tabela 7 – Dados utilizados para gerar as instâncias de teste para CGBH e PIPSAPAB-I.

Parâmetro	Descrição
Rotas	Foram consideradas 90 rotas.
Períodos e Microperíodos	A duração máxima de cada período (H_t) é de 11 horas, para respeitar as restrições da maré alta, e também considerar um tempo razoável para atracação e desatracação dos navios. Cada microperíodo tem a duração limitado em 11 horas (μ_{max}).
Berços	O terminal apresenta três berços. No qual os berços apresentam, respectivamente, os comprimentos (sb_b) de 300, 330, 360 metros.
Fila de navios	O comprimento dos navios (sv_n) que são carregados no terminal, está dividido em três casos: 50% dos navios apresentam comprimento inferior a 300 metros, 40% dos navios apresentam comprimento superior a 300 metros e inferior a 330 metros, 10% dos navios apresentam comprimento superior a 330 metros.

Para a heurística de mergulho com *backtracking* da CGBH (DHB-I,) os parâmetros utilizados são: $maxdepth = 8$, $\Omega = 0.99$ e $\epsilon = 0.01$. Por meio desses parâmetros pretende-se fixar menos variáveis em cada nó e realizar mais vezes *backtracking*.

Nas tabelas apresentadas a seguir, a coluna *ins.* representa o número da instância, a coluna *prod.* representa o número de produtos manipulados no terminal, a coluna *per.* representa a quantidade de períodos que o horizonte de planejamento apresenta, e a coluna *nav.* representa a quantidade de navios atendidos durante o horizonte de planejamento. As colunas f e f_{ub} fornecem os valores dos melhores limites superiores obtidos pelo

algoritmo *branch-and-cut* do *solver* CPLEX e pela heurística baseada em geração de coluna, respectivamente. A coluna f_{lb_1} fornece o limite inferior para o MIP obtido por sua relaxação linear. A coluna f_{lb_2} fornece o valor da solução ótima do PMLR obtida após a convergência do procedimento de geração de coluna. A coluna GAP_1 fornece o GAP entre o limite superior e inferior obtidos pelo algoritmo do *solver* CPLEX. A coluna GAP_2 fornece o GAP relativo entre os limites superiores do *solver* CPLEX e da heurística CGBH, dado $GAP_2 = ((f_{ub}/f) - 1)100$. A coluna GAP_3 fornece o GAP : $GAP_3 = ((f_{ub}/f_{lb_2}) - 1)100$. Finalmente, as colunas t_{cplex} e t_{heu} são os tempos computacionais decorridos, expressos em segundos.

A Tabela 8 apresenta resultados computacionais para um conjunto de instâncias com variação de oferta, demanda e estoque inicial. As instâncias foram geradas artificialmente para os cenários de oferta e demanda baixa e alta. No cenário de oferta e demanda baixa, o volume de carga e a quantidade de navios disponíveis são inferiores em relação ao cenário de alta. Simultaneamente, são investigadas três diferentes situações de estoque inicial. Para dividir as instâncias foram adicionadas as colunas *conj.* e *tipo*. Na coluna *conj.*, o conjunto 1 representa as instâncias com oferta e demanda baixa e o conjunto 2 representa as instâncias com oferta e demanda alta. A coluna *tipo* representa a porcentagem de uso do estoque no início do horizonte.

Os resultados da Tabela 8 mostram que o *solver* CPLEX pode produzir resultados para apenas certas instâncias de pequeno porte, gerando um limite superior para apenas 30% das instâncias. Sendo que, nenhuma dessas instâncias foram solucionadas otimamente. Nas demais instâncias, o *solver* não conseguiu produzir nenhum resultado significativo devido à memória insuficiente.

Com a heurística CGBH foi possível obter limites superiores para todas as instâncias. Os valores da função objetivo indicam que todos os suprimentos e carregamentos são atendidos, demonstrando uma característica de soluções de qualidade. Os baixos valores da coluna GAP_3 também apontam boa qualidade das soluções geradas pela heurística para aquelas instâncias que o *solver* não conseguiu encontrar resultados. Ao comparar as soluções dos dois métodos, quando possível, o *solver* apresenta soluções de melhor qualidade, no entanto o tempo decorrido é muito superior ao da heurística CGBH, que apresenta uma solução e um tempo computacional satisfatório. Destaque para a instância 04, em que a heurística CGBH encontra um limite superior melhor do que o obtido via *solver*.

Em relação aos tempos computacionais, o aumento na quantidade de produtos manipulados, em especial no número de períodos e navios, e na demanda tornam o problema integrado ainda mais complexo e conseqüentemente requerem mais tempo para obter soluções. Uma explicação para este comportamento, no ponto de vista do terminal,

está relacionada à necessidade de alocar mais subáreas para estocar os produtos, ativar mais rotas, sequenciar mais navios, dentre outras mais explicações. No ponto de vista da formulação matemática, o número de restrições e variáveis aumentam da mesma maneira que crescem os parâmetros, tornando assim a solução do PMLR mais difícil.

Tabela 8 – Resultados computacionais para um conjunto de instâncias com variação oferta, demanda e estoque inicial.

ins.	conj.	tipo	per.	nav.	prod.	CPLEX				Heurística baseada na geração de coluna				
						f_{lb_1}	f	t_{cplex}	GAP_1	f_{lb_2}	f_{ub}	t_{heu}	GAP_2	GAP_3
01	1	estoq.	10	15	10	500.26	524.10	21600	0.54	500.26	530.91	1678	1.30	6.12
02		em 10%			15	493.05	512.98	6655(-)	2.30	493.05	527.47	1532	2.82	6.98
03					20	514.74	-	-	-	514.74	545.47	2570	-	5.96
04			15	23	15	760.71	853.27	6275(-)	9.83	760.71	800.43	5616	-6.19	5.22
05					20	766.89	-	-	-	766.89	801.09	9557	-	4.46
06		estoq.	10	15	10	497.39	516.63	21600	1.02	497.39	520.76	2141	0.80	4.69
07		em 50%			15	500.07	-	-	-	500.07	525.97	2010	-	5.17
08					20	513.66	-	-	-	513.66	540.38	4542	-	5.20
09			15	23	15	763.45	-	-	-	763.45	804.78	4987	-	5.41
10					20	769.72	-	-	-	769.72	809.01	9608	-	5.10
11		estoq.	10	15	10	501.88	513.85	21600	0.41	501.88	523.57	423	1.89	4.32
12		em 80%			15	496.88	512.39	12961(-)	1.57	496.88	517.92	1378	1.08	4.23
13					20	512.44	-	-	-	512.44	542.79	2142	-	5.92
14			15	23	15	761.22	-	-	-	761.22	797.11	5357	-	4.71
15					20	765.05	-	-	-	765.05	811.31	8592	-	6.04
16	2	estoq.	10	18	10	522.68	534.31	21600	1.16	522.68	548.29	1176	2.62	4.89
17		em 10%			15	527.67	-	-	-	527.67	555.89	3465	-	5.34
18					20	534.71	-	-	-	534.71	563.08	2737	-	5.30
19			15	25	15	770.57	-	-	-	770.57	809.58	5423	-	5.06
20					20	781.44	-	-	-	781.44	829.91	9221	-	6.20
21		estoq.	10	18	10	521.23	532.53	21600	0.85	521.23	540.21	443	1.44	3.63
22		em 50%			15	528.96	-	-	-	528.96	560.68	3437	-	5.99
23					20	533.92	-	-	-	533.92	566.57	6153	-	6.11
24			15	25	15	774.09	-	-	-	774.09	811.37	5158	-	4.81
25					20	774.60	-	-	-	774.60	813.31	8406	-	4.99
26		estoq.	10	18	10	524.49	540.39	21600	1.58	525.49	549.36	946	1.66	4.54
27		em 80%			15	531.00	-	-	-	531.00	566.88	3483	-	6.75
28					20	534.66	-	-	-	534.66	559.49	3503	-	4.64
29			15	25	15	778.17	-	-	-	778.17	821.86	5132	-	5.61
30					20	807.25	-	-	-	807.25	834.03	8116	-	3.31

⁽⁻⁾ refere-se instâncias em que o solver não conseguiu obter uma solução para o problema integrado por falta de memória.

Para as instâncias das Tabelas 9, 10 e 11, o estoque inicial de produtos nos pátios permanece em 30% da capacidade total do pátio.

Para a Tabela 9 foram construídos três conjuntos de instâncias: no conjunto 1, foi considerado que qualquer um dos P produtos disponíveis pode ser alocado em qualquer uma das subáreas. No conjunto 2, o total de produtos foi dividido em dois subconjuntos: o primeiro subconjunto de produtos pode ficar armazenado somente nas subáreas associadas

aos pátios 1 e 2, e o segundo subconjunto de produtos, apenas nas subáreas associadas ao pátio 3, 4 e 5. No conjunto 3, os P produtos foram divididos em 5 subconjuntos, e atribuídos para cada um dos 5 pátios. A coluna *conj.* representa o conjunto das instâncias.

Tabela 9 – Resultados computacionais comparando três conjuntos de instâncias com variação na alocação de produtos nas subáreas.

ins.	conj.	per.	nav.	prod.	CPLEX				Heurística baseada na geração de coluna				
					f_{lb_1}	f	t_{cplex}	GAP_1	f_{lb_2}	f_{ub}	t_{heu}	GAP_2	GAP_3
31	1	10	15	5	471.36	487,87	1969(-)	2.36	471.36	495.05	153	1.47	5.02
32				10	471.07	4226,27	3555(-)	88.61	471.07	495.95	364	-88.27	5.28
33				15	471.69	-	-	-	471.69	492.75	3078	-	4.44
34				20	506.70	-	-	-	506.70	531.62	3501	-	4.91
35		15	22	15	705.82	-	-	-	705.82	740.93	4906	-	4.97
36				20	721.89	-	-	-	721.89	762.38	10779	-	5.61
37	2	10	15	5	477.18	485,60	2839(-)	1.07	477.18	491.47	689	1.21	2.99
38				10	477.35	-	-	-	477.35	490.61	530	-	2.77
39				15	477.91	-	-	-	477.91	493.47	1977	-	3.25
40				20	493.94	-	-	-	493.94	512.30	1692	-	3.71
41		15	22	15	718.88	-	-	-	718.88	739.10	4120	-	2.81
42				20	728.81	-	-	-	728.81	754.86	5610	-	3.57
43	3	10	15	5	483.14	495,49	2803(-)	2.49	483.14	488.07	59	-1.50	1.02
44				10	484.17	-	-	-	484.17	490.53	1021	-	1.31
45				15	484.31	-	-	-	484.31	489.17	532	-	1.00
46				20	510.27	-	-	-	510.27	518.27	987	-	1.53
47		15	22	15	729.53	-	-	-	729.53	739.96	2209	-	1.43
48				20	748.33	-	-	-	748.33	763.14	3175	-	1.97

⁽⁻⁾ refere-se instâncias em que o solver não conseguiu obter uma solução para o problema integrado por falta de memória.

Os resultados apresentados na Tabela 9 demonstram novamente a dificuldade do *solver* para produzir uma solução para o problema integrado proposto. Apenas 22% das instâncias o solver conseguiu produzir um limite superior. Para as demais instâncias, o solver foi incapaz de produzir um limite considerável por falta de memória. Quando considerada a heurística CGBH, os resultados são similares aos encontrados na Tabela 8, no qual a mesma conseguiu gerar bons limites superiores para todas as instâncias.

Nas instâncias 31, 32, 37 e 43, é possível comparar as soluções geradas pelos dois métodos (GAP_2), e assim notar que as soluções dadas pela heurística se aproximam mais do limite superior dado pelo *solver* (em uma instância obtém até um melhor) nos conjunto que a faixa de produtos disponíveis para armazenamento é reduzida. Quando, em cada subárea, a faixa de produtos disponíveis para armazenamento é reduzida, a complexidade e o tamanho dos problemas são também reduzidas. Nota-se também uma variação nos valores das funções objetivo das soluções. Uma explicação para este comportamento são justamente

as restrições de alocação de produtos nas subáreas. Por meio destas restrições, torna-se necessário transportar os produtos por rotas mais caras, consequentemente elevando os valores de custo.

A Tabela 10 apresenta resultados da desempenho do *solver* e da heurística CGBH para instâncias com variações no número de navios atendidos pelo terminal para diferentes quantidades de períodos e produtos manipulados. As instâncias estão separadas em três conjuntos, o que representa três situações de congestionamento no terminal. O congestionamento está associado à quantidade de navios que devem ser carregados e o número de períodos no horizonte de planejamento. Quanto maior o número de navios para um mesmo número de período, maior é o congestionamento. Por exemplo, na instância 49 tem-se 13 navios para serem carregados em 5 períodos, já a instância 55 possui 15 navios que para serem atendidos em 10 períodos, logo a instância 49 apresenta um congestionamento inferior ao da instância 55. Os conjuntos 1, 2 e 3 representam, respectivamente, o congestionamento baixo, médio e alto.

Tabela 10 – Resultados computacionais para um conjunto de instâncias com variação no número de navios.

ins.	conj.	nav.	per.	CPLEX					Heurística baseada na geração de coluna				
				prod.	f_{lb_1}	f	t_{cplex}	GAP_1	f_{lb_2}	f_{ub}	t_{heu}	GAP_2	GAP_3
49	1	13	10	10	482.82	503.12	17917(-)	0.64	482.82	515.29	477	2.42	6.72
50				15	504.75	-	-	-	504.75	536.41	1126	-	6.27
51		21	15	10	731.32	-	-	-	731.32	765.27	3010	-	4.64
52				15	740.72	-	-	-	740.72	781.06	5331	-	5.44
53		28	20	10	981.80	-	-	-	981.80	1028.56	4785	-	4.78
54				15	984.44	-	-	-	984.44	1022.89	9618	-	3.90
55	2	15	10	10	494.79	509.72	21600	1.07	494.79	520.85	553	2.18	5.26
56				15	492.45	-	-	-	492.45	515.81	1346	-	4.74
57		23	15	10	751.77	-	-	-	751.77	791.08	3220	-	5.22
58				15	751.57	-	-	-	751.57	792.31	5755	-	5.42
59		30	20	10	1002.71	-	-	-	1002.71	1064.77	5080	-	6.19
60				15	998.85	-	-	-	998.85	1050.61	9836	-	5.18
61	3	17	10	10	510.58	526.39	21600	1.06	510.58	536.64	2135	1.95	5.10
62				15	533.26	-	-	-	533.26	564.98	1624	-	5.94
63		25	15	10	765.33	-	-	-	765.33	804.53	3307	-	5.12
64				15	765.28	-	-	-	765.28	799.31	5928	-	4.44
65		32	20	10	1020.96	-	-	-	1020.96	1071.81	5334	-	4.98
66				15	1018.16	-	-	-	1018.16	1082.96	10403	-	6.36

⁽⁻⁾ refere-se instâncias em que o *solver* não conseguiu obter uma solução para o problema integrado por falta de memória.

Uma análise dos resultados da Tabela 10 indicam que conforme as instâncias se tornam mais complexas, em especial com aumento do número de navios e períodos, mais difícil é encontrar uma solução para o problema integrado em questão. Para apenas 16% das instâncias, o *solver* encontrou um limite superior com um GAP inferior a aproximadamente

1%, evidenciando mais uma vez a ineficiência do *solver* em solucionar o problema estudado. Quando considerada a heurística CGBH, esse aumento de dificuldade foi expresso no crescimento gradual dos tempos computacionais.

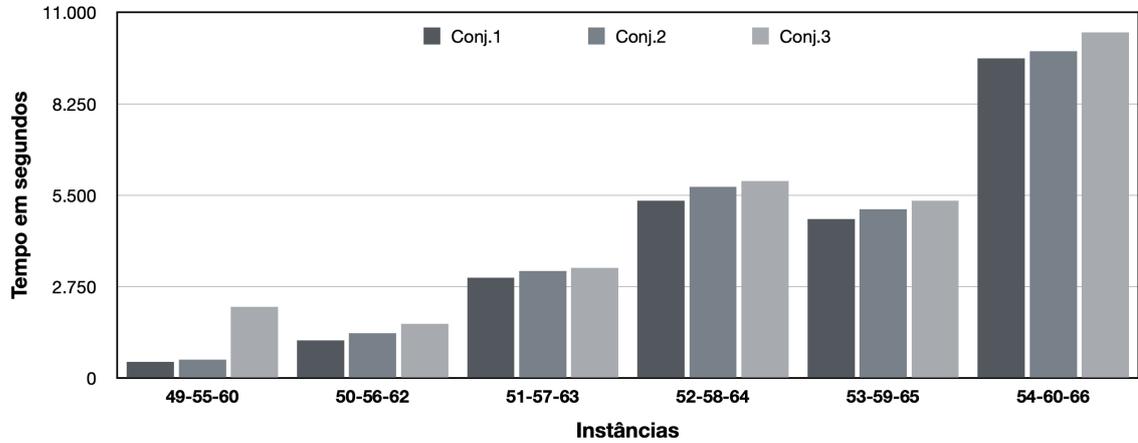


Figura 15 – Comparação dos tempos computacionais entre os três conjuntos.

A Figura 15 apresenta uma comparação entre os tempos computacionais da heurística CGBH para os três conjuntos de instâncias. Verifica-se uma expressiva diferença de tempo entre as instâncias 49 e 66. Uma explicação para esse comportamento está relacionada ao PMLR e ao procedimento de geração de coluna. Em instâncias maiores, o PMLR requer um tempo maior para obter uma solução, e o procedimento de geração de coluna deve ser realizado mais vezes para gerar mais colunas com microperíodos para o PMLR, pois nestas instâncias as rotas estarão transportando um volume maior de produtos, e consequentemente mais microperíodos serão necessários. Esta justificativa pode ser também considerada explicar as variações dos tempos computacionais as demais tabelas.

Em relação aos resultados da CGBH, novamente a heurística se mostrou eficiente em obter limites superiores de qualidade para todas as instâncias, como mostram os valores da coluna GAP_3 . Com destaque para as instâncias 49, 55 e 61, que apresentaram um GAP inferior a 3% em relação ao limite superior obtido pelo *solver*.

A Tabela 11 apresenta os resultados computacionais para instâncias com variação no número de berços. Estas instâncias foram divididas em dois conjuntos: no primeiro conjunto, considerou a mesma configuração utilizada nas instâncias das tabelas anteriores. E no segundo conjunto, foi considerado 4 berços de atracação, no qual adicionou mais um berço com comprimento intermediário. Na tabela em questão, a quantidade de produtos manipulados foi fixada em 10. A coluna *ber.* representa o número de berços de atracações considerados pelas instâncias.

Tabela 11 – Resultados computacionais para um conjunto de instâncias com variação no número de berços

ins.	ber.	nav.	per.	CPLEX				Heurística baseada na geração de coluna				
				f_{lb_1}	f	t_{cplex}	GAP_1	f_{lb_2}	f_{ub}	t_{heu}	GAP_2	GAP_3
67	3	10	8	339.65	349.09	13031	0.00	339.65	359.31	118	2.93	5.78
68		15	10	492.73	504.61	21600	0.42	492.73	518.55	1372	2.76	5.24
69		20	13	650.64	685.89	21600	3.80	650.64	681.26	2732	-0.68	4.70
70		25	17	843.09	-	-	-	843.09	890,71	3311	-	5.64
71		30	20	999.99	-	-	-	999.99	1031,11	4906	-	3.11
72	4	10	6	335.92	390,02	21600	2.01	335.92	410,02	342	5,13	22.05
73		15	8	553.24	-	-	-	553.24	684,55	1202	-	23.73
74		20	10	706.50	-	-	-	706.50	864,24	3991	-	22.32
75		25	14	923.64	-	-	-	923.64	1132,43	6359	-	22.60
76		30	16	1142.91	-	-	-	1142.91	1391,19	10041	-	21.72

^(c) refere-se instâncias em que o solver não conseguiu obter uma solução para o problema integrado por falta de memória.

Considerando as instâncias com três berços, o *solver* obteve soluções para 60% dos experimentos, sendo uma delas uma ótima. Enquanto para instâncias apresentando quatro berços, apenas 10% dos experimentos. Portanto, a adição de mais um berço tornou o problema ainda mais difícil, no ponto de vista do *solver*. Em relação a heurística CGBH, mais uma vez a mesma gerou soluções para todas as instâncias, no entanto é notável a diferença dos tempos para os dois conjuntos de instância. Na instância 67 e 68, a heurística precisou de um tempo muito menor para apresentar uma solução muito próxima das obtidas pelo *solver*. E ainda mais, na instância 68, a CGBH obteve uma solução de melhor qualidade (GAP_2) do que a encontrada pelo *solver*, em um tempo oito vezes inferior.

Na primeira instância com quatro berços, nota-se que o *gap* entre as soluções dos dois algoritmos é superior em relação aos valores obtidos nas instâncias com três berços. Uma explicação para isso, está justamente relacionado a adição de mais um berço. Este incremento permitiu à DHB-I obter soluções com valores de custos próximos, porém muito distintas. No qual a solução selecionada (a qual apresenta menor valor de custo) pode gerar ao final da CGBH uma solução não tão boa quanto uma outra solução com valor de custo superior muito próxima. No entanto, para verificar se essa segunda opção pode gerar uma solução melhor é um procedimento muito custoso computacionalmente, pois o processo de fixação deve ser repetido duas vezes. Ao final, a melhor solução pode ser dada pela primeira opção, como ocorre na maioria dos casos.

Por fim, a Tabela 12 apresenta os resultados computacionais com variação no número de rotas. Mais uma vez, as instâncias são divididas em dois conjuntos: no conjunto 1, são consideradas as mesmas 90 rotas utilizadas nas tabelas anteriores, e no conjunto 2, o

terminal apresenta 150 rotas. Para estas instâncias foram considerados 3 berços com a mesma configuração inicial, o estoque inicial de produtos no pátio permaneceu em 50% da capacidade do pátio e os parâmetros restantes são os mesmo das instâncias anteriores. A coluna *rotas* representa o número de rotas transportadoras de produtos considerados.

Tabela 12 – Resultados computacionais para um conjunto de instâncias com variação no números de rotas disponíveis.

ins.	rotas	prod.	per.	nav.	CPLEX				Heurística baseada na geração de coluna				
					f_{lb_1}	f	t_{cplex}	GAP_1	f_{lb_2}	f_{ub}	t_{heu}	GAP_2	GAP_3
77	90	5	10	12	438.59	444,49	2011(-)	0.52	438.59	458,22	86	3,09	4.45
78				14	446.87	466,66	3303(-)	2.89	446.87	469,15	108	0,53	4.98
79			15	18	659.26	678,60	4006(-)	2.17	659.26	681,59	1365	0,44	3.38
80				20	668.85	-	-	-	668.85	701,71	1465	-	4.91
81		10	15	18	624.55	-	-	-	624.55	643,25	2175	-	2.99
82				20	633.06	-	-	-	633.06	655,25	3151	-	3.50
83	150	5	10	12	401.60	423,26	3977(-)	3.31	401.60	443,57	220	4,80	10.49
84				14	412.30	427,22	9555(-)	1.54	412.30	467,34	276	9,39	13.34
85			15	18	598.68	-	-	-	598.68	681,34	1522	-	13.80
86				20	617.85	-	-	-	617.85	682,89	1510	-	10.52
87		10	15	18	577.21	-	-	-	577.21	685,43	2936	-	18.74
88				20	608.22	-	-	-	608.22	696,47	3889	-	14.51

⁽⁻⁾ refere-se instâncias em que o solver não conseguiu obter uma solução para o problema integrado por falta de memória.

Nos experimentos descritos na Tabela 12, a heurística CGHB gerou limites superiores para todas os experimentos. Porém os valores de GAP_2 encontrado nas instâncias 83 e 84 do segundo conjunto são superiores aos 3% obtido nas instâncias 77, 78 e 79 do primeiro conjunto. Além disso, os valores de GAP_3 também maiores nas instâncias com 150 rotas. Em relação ao solver, o CPLEX obteve limites superiores para apenas 40% dos experimentos, sendo que nesses casos não foi possível encontrar uma melhor solução devida a necessidade de mais memória. Nota-se então que a ampliação no número de rotas torna o problema ainda mais complicado. A razão disto está relacionada ao aumento da complexidade do sequenciamento. Com esta expansão, mais rotas estarão compartilhando equipamentos, como consequência mais complicado é selecionar as rotas com os menores custos.

Um comportamento observado em todas as tabelas apresentadas foi a variação dos valores da função objetivo. Uma explicação para esta variação está relacionada ao custo de manutenção (σ^r) das rotas. Em razão dos aumentos da quantidade de períodos, dos navios e dos produtos manipulados, maiores fluxos de produtos escoam pelas rotas, e assim superiores serão os custos para manutenção.

6.3 Resultados para algoritmo CMH na solução do PIPSAPAB-II

Para os experimentos computacionais da AH na solução do *PIPSAPAB-II*, além das apresentados anteriormente, são também considerados os seguintes parâmetros.

Tabela 13 – Dados utilizados para gerar as instâncias de teste para o algoritmo CMH e *PIPSAPAB-II*.

Parâmetro	Descrição
Rotas	Foram consideradas 150 rotas.
Períodos e Microperíodos	A duração máxima de cada período (H_t) é de 11 horas, para respeitar as restrições da maré alta, e também considerar um tempo razoável para atracação e desatracação dos navios. Todos os períodos serão divididos em 11 microperíodos, logo a duração de cada microperíodo (μ_{it}) é de 1 hora.
Berços	O terminal apresenta três berços. No qual os berços apresentam, respectivamente, os comprimentos (sb_b) de 335, 300, 240 metros
Fila de navios	O comprimento dos navios (sv_n) que são atendidos pelo terminal, está dividido em três casos: 10% são navios possuem um comprimento superior a 315 metros, 40% dos navios possuem um comprimento entre 275 e 290 metros e 50% dos navios possuem comprimento inferior a 240 metros.

Ao comparar os dados apresentados na Tabelas 7 e 13, note-se que os comprimentos dos navios foram reduzidos, logo o volume de produto demandados pelos navios são inferiores. Outro dado alterado foi o número de rotas, que foi ampliado para 150 rotas consideradas. Essas modificações foram realizadas em detrimento do sequenciamento sem preempção do PIPSAPAB-II. A Tabela 14 compara os limites inferiores gerado pela solução dos PIPSAPAB-I (coluna f_{lb_I}) e PIPSAPAB-II (coluna $f_{lb_{II}}$) relaxados para conjunto de instâncias apresentados na Tabela 9. Em uma análise das duas colunas, verifica-se que os limites inferiores obtidos pelo PIPSAPAB-II são maiores em relação do PIPSAPAB-I. A explicação para as diferenças de valores está relacionada ao não atendimento das ofertas e demandas. Os dois primeiros termos da função objetivo são responsáveis por contabilizar o não atendimento, como também são os únicos capazes de gerar essa expressiva diferença de valores. Esse comportamento pode ser interpretado como uma redução na capacidade das rotas de transportar fluxos de produtos. Portanto, o aumento na quantidade de rotas é atribuído, para ampliar as possibilidades de caminho que fluxo de produtos possam ser transportados, sem que necessite que alterar a quantidade de equipamentos. Além disso, mesmo com a redução no comprimento dos navios as instâncias permanecem próximas da realidade.

Outra alteração feita nas instâncias desta seção em relação a seção anterior (6.2), é a redução no número de navios disponíveis no horizonte de planejamento. A justificativa para essa consideração está associada a Tabela 14 e a redução na capacidade das rotas de transportar fluxos de produtos. Desse modo, as instâncias geradas para esta seção continuam baseadas na realidade de um terminal, porém apresentam uma pequena redução na escala em relação da seção anterior.

Tabela 14 – Comparação de limites inferiores entre PIPSAPAB-I e PIPSAPAB-II.

ins.	conj.	per.	nav.	prod.	f_{lb_I}	$f_{ub_{II}}$
31	1	10	15	5	471.36	25325.83
32				10	471.07	27430.20
33				15	471.69	32732.89
34				20	506.70	37343.76
35		15	22	15	705.82	50487.83
36				20	721.89	64504.32

Para a heurística de mergulho com backtracking do algoritmo CMH os parâmetros utilizados são: $maxdepth = 5$, $delta = 0.95$ e $step = 0.05$. Por meio desses parâmetros pretende-se fixar uma quantidade razoável variáveis em cada nó para a árvore não ficar muito extensa e que o tempo computacional seja reduzido, mas ao mesmo tempo gerar boas soluções inteiras para as variáveis w_{nt}^b . Para o *local branching* das variáveis v_{nt}^b os parâmetros são: $it_{max} = 5$, $t_{limit} = [180, 540]$ segundos e $k_0 = \lfloor \frac{|T|}{2} \rfloor$. Enquanto que, para o *local branching* das variáveis v_{nt}^b os parâmetros são: $it_{max} = 2$, $t_{limit} = [900, 1200]$ segundos e $k_0 = |T|$. Esses valores foram obtidos após inúmeros testes e notou-se que com esses valores houve um maior benefício entre tempo computacional e limite superior encontrado. A variação nos valores do parâmetro em ambos casos está associada à quantidade de tipo de produtos manipulados.

Na tabela apresentada nesta seção, a coluna *ins.* representa o número da instância, a coluna *prod.* representa o número de produtos manipulados no terminal, a coluna *per.* representa a quantidade de períodos que o horizonte de planejamento apresenta, e a coluna *nav.* representa a quantidade de navios atendidos durante o horizonte de planejamento. As colunas f e f_{ub} fornecem os valores melhores limites superiores obtidas pelo algoritmo *branch-and-cut* do *solver CPLEX* e pelo algoritmo CMH, respectivamente. A coluna f_{lb} fornece o limite inferior para o MIP obtido por sua relaxação linear. A coluna GAP_1 fornece o *GAP* entre o limite superior e inferior obtidos pelo algoritmo do *solver CPLEX*. A coluna GAP_2 fornece o *GAP* relativo entre a melhor solução inteira do *solver CPLEX* e da heurística, dado $GAP_2 = ((f_{ub}/f) - 1)100$. A coluna GAP_3 fornece o *GAP*: $GAP_3 = ((f_{ub}/f_{lb}) - 1)100$. Finalmente, as colunas t_{cplex} e t_{heu} são os tempos computacionais decorridos, expressos em segundos.

Análogo a Tabela 10, a Tabela 15 exibe os resultados computacionais para um conjunto de instâncias que possuem variações do número de navios disponíveis para o carregamento durante todo horizonte de planejamento. Logo, as instâncias estão divididas em congestionamento baixo, médio e alto.

Tabela 15 – Resultados computacionais para um conjunto de instâncias com variação no número de navios.

ins.	conj.	nav.	per.	prod.	CPLEX				Algoritmo CMH			
					f_{lb}	f	t_{cplex}	GAP_1	f_{ub}	t_{heu}	GAP_2	GAP_3
89	1	3	5	5	52.19	54.49	839	0.00	54.49	276	0,00	4.40
90				10	51.29	55.29	2456	0.00	56.41	1170	2,03	9.97
91		8	10	5	125.63	139.62	21600	4.33	146.37	2702	4,84	16.51
92				10	127.42	154.04	21600	9.40	153.33	4272	-0,46	20.33
93		12	15	5	196.54	237.34	21600	10.70	241.00	3558	1,54	22.61
94				10	199.57	-	-	-	266.55	7777	-	33.55
95	2	5	5	5	75.66	77.30	658	0.00	78.60	267	1,68	3.89
96				10	75.85	81.38	2851	0.00	82.19	1416	0,98	8.35
97		10	10	5	155.66	172.25	21600	4.56	177.98	2245	3,33	14.33
98				10	156.70	218.25	21600	23.62	182.55	4926	-16,36	16.45
99		15	15	5	234.06	273.20	21600	11.02	284.45	4028	4,12	21.52
100				10	232.67	-	-	-	310.58	9599	-	33.48
101	3	7	5	5	101.89	103.26	626	0.00	103.67	166	0,39	1.74
102				10	103.49	110.77	6794	0.00	114.04	1389	2,95	10.19
103		12	10	5	180.50	206.90	21600	7.14	208.21	2827	0,64	15.35
104				10	183.81	282.52	21600	31.18	241.66	4609	-14,46	31.47
105		17	15	5	255.66	311.47	21600	13.38	294.76	5487	-5,36	15.29
106				10	254.09	-	-	-	351.46	10220	-	38.32

^(*c*) refere-se instâncias em que o solver não conseguiu obter uma solução para o problema integrado por falta de memória.

Os resultados apresentados na tabela demonstram a grande dificuldade do *solver* CPLEX em encontrar boas soluções para o MIP. Em apenas 33 % das instâncias o *solver* obteve uma solução ótima, sendo essas consideradas as instâncias de menor porte. Enquanto que em 50% das instâncias o *solver* foi apenas capaz de encontrar um limite superior dentro das 6 horas limites de teste. Nos restantes 17%, o CPLEX não conseguiu produzir nenhum resultado significativo devido à memória insuficiente.

Com o algoritmo CMH foi possível obter bons limites superiores para todas as instâncias, e em todas soluções todas as ofertas e demandas foram atendidas. Nas instâncias solucionadas otimamente pelo *solver* (instâncias 105, 106, 111, 112, 117 e 118), o algoritmo obteve limites superiores muito próximos da solução ótima, tendo GAPs inferior a 2%. Com destaque para instância 01, que o algoritmo obteve solução ótima em um terço do tempo computacional do *solver*. Ao comparar os soluções entre os dois métodos, através das colunas GAP_2 , t_{cplex} e t_{heu} , o solver obtém um melhor limite superior ao do algoritmo em 55% das instâncias, porém o tempo computacional do *solver* é significativamente superior ao do algoritmo. Note que, em 23% das instâncias o algoritmo encontrou um limite superior com custo inferior do encontrado pelo *solver*. Por fim, os baixos valores

encontras na coluna GAP_3 apontam a boa qualidade dos limites superiores encontrados pelo algoritmo.

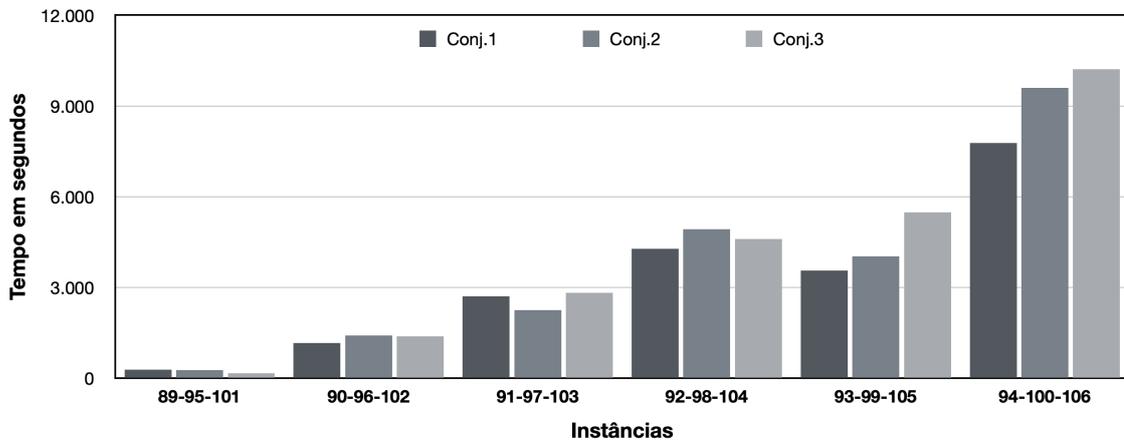


Figura 16 – Comparação dos tempos computacionais entre os três conjuntos.

A Figura 16 apresenta uma comparação entre os tempos computacionais para os três conjuntos de instâncias. Quando analisada separadamente cada barra, verifica-se o crescimento dos tempos computacionais de acordo que aumenta a quantidade de tipos de produtos manipulados, o número de períodos e fila de navios. Esse comportamento é esperado, uma vez que o problema integrado torna-se mais difícil. Mesma a solução da formulação relaxada apresenta um tempo computacional maior. A maior quantidade de tipos de produtos manipulados e de períodos requerem um melhor gerenciamento de produtos nos pátios, pois as subáreas podem alocar apenas um produto e os custos de trocas devem ser minimizados, como consequência exigem um maior tempo e melhor desempenho da $LB-F$ em obter soluções em que toda a oferta seja atendida. Além disso, o valor do parâmetro (k_0) da $LB-F$, é responsável em determinar o espaço mínimo de busca está associado ao número de períodos, logo com espaço maior é necessário um tempo maior para analisar. Outro detalhe da $LB-F$, é que o parâmetro t_{limit} está associado com a quantidade de tipos de produtos manipulados. As heurísticas HRF-F e HRF-V também ajudam a ampliação do tempo computacional, pois acréscimos de períodos geram mais iterações para as heurísticas. Como as operações estão integradas, um horizonte de planejamento mais longo afeta o problema de sequenciamento, pois cresce a quantidade de rotas diferentes ativadas e número de microperíodos alocados. O aumento no tempo também está associado a HRH, pois essa heurística está fortemente relacionada com as duas operações comentadas e tem o número de iterações igual ao de períodos.

As diferenças dos tempos computacionais entre as três cores de barras também está relacionada à complexidade. No entanto, a dificuldade está associada à variação na quantidade de navios atendidos para um mesmo número de período. Uma justificativa para o comportamento das barras que representa o conjunto 3 de instâncias, está relacionada à duração da heurística de mergulho com backtracking e a $LB-V$. Com relação a heurística

de mergulho com backtracking, a maior quantidade de navios origina um maior espaço de solução e mais ramificações para a árvore. Enquanto que para a LB-V, o parâmetro de distância mínima (k_0) está associada ao número de navios, logo quanto maior o valor de k_0 maior o espaço de busca e tempo gasto.

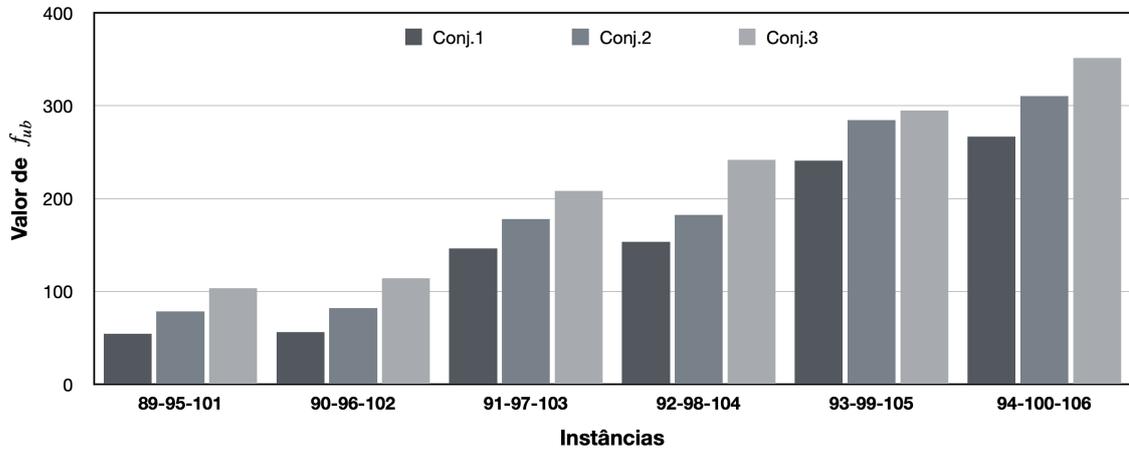


Figura 17 – Comparação dos valores de f_{ub} entre os três conjuntos.

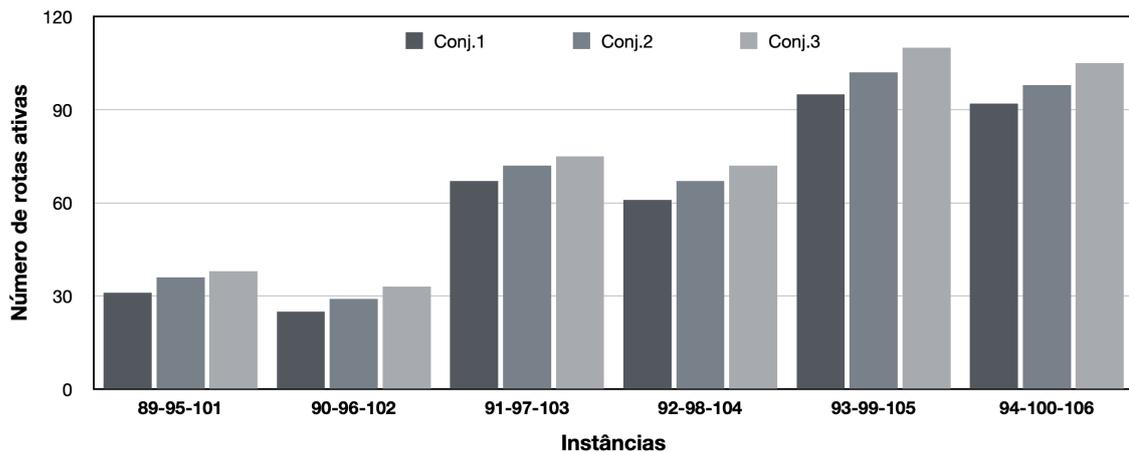


Figura 18 – Comparação do número de rotas utilizadas entre os três conjuntos.

Do mesmo modo que os resultados obtidos pela CGBH, a variação nos valores de f obtido via o algoritmo CMH está associado ao custo de manutenção das rotas (σ^r) e as demandas dos navios. A Figura 17 apresenta uma comparação entre os valores da coluna f_{ub} para os três conjuntos de instâncias. Com o aumento da quantidade de períodos, de navios e de produtos manipulados, têm-se maiores fluxos de produtos escoando pelas rotas, conseqüentemente os custos de manutenção das rotas serão superiores. A Figura 18, que compara o número de rotas utilizadas durante todo horizonte de planejamento, exibe o crescimento de rotas ativas de acordo com o aumento do fluxo de produtos. Além disso, o valor da função objetivo está associado à quantidade de navios atendidos, então quanto maior a quantidade de navios atendidos no horizonte de planejamento, maior será o valor do último termo de f .

Para estudar a influência do *local branching* no algoritmo CMH, foram realizados experimentos com o algoritmo sem a heurística *local branching*. Os experimentos utilizaram os conjuntos 1 e 3 de instâncias da Tabela 15. A Tabela 16 apresenta e compara os resultados computacionais obtidos. Nesta tabela, a coluna f'_{ub} está associada ao limites superiores obtidos pelo algoritmo CMH sem *local branching*.

Tabela 16 – Resultados computacionais comparando o algoritmo CMH com e sem *local branching*.

ins.	conj.	nav.	per.	prod.	Algoritmo CMH	Algoritmo CMH
					com <i>local branching</i>	sem <i>local branching</i>
					f_{ub}	f'_{ub}
89	1	3	5	5	54.49	84.13
90				10	56.41	95.88
91		8	10	5	146.37	229.08
92				10	153.33	24076.38
93		12	15	5	241.00	391.89
94				10	266.55	59263.52
101	3	7	5	5	103.67	157.30
102				10	114.04	194.08
103		12	10	5	208.21	302.63
104				10	241.66	46938.75
105		17	15	5	294.76	457.73
106				10	351.46	105924.13

Ao analisar os resultados dos experimentos da Tabela 16, torna-se evidente que utilizar as heurísticas *local branching* melhora consideravelmente a eficácia do algoritmo. A última coluna da Tabela 16 destaca a diferença entre os limites superiores. Esse resultado era esperado, já que as heurísticas HRF-V e HRF-F são consideradas gulosas, e o objetivo da implementação das heurísticas *local branching* é refinar as soluções. Quando analisado os limites superiores do algoritmo sem *local branching*, verifica-se que uma parte das instâncias (89, 90, 91, 93, 101, 102, 103, 105) não apresentam uma boa qualidade, mas são soluções em que toda oferta e demanda é atendida. Enquanto que, na outra parte das instâncias (92, 94, 104 e 106), os limites superiores são muito altos, demonstrando que uma parte da oferta e demanda não é atendida. Mas os bons limites superiores obtidos com aplicação da heurística *local branching*, tem em contrapartida consideráveis diferenças de tempo de execução do algoritmo.

Capítulo 7

Conclusão e trabalhos futuros

A conclusão da pesquisa e os trabalhos futuros são apresentados neste capítulo. A Seção 7.1 apresenta um sumário do trabalho realizado. A Seção 7.2 discute possíveis linhas de pesquisa para continuar este estudo.

7.1 Conclusão

Nesta dissertação, foi estudado um problema integrado de planejamento, sequenciamento, alocação de pátio e alocação de berço para um terminal portuário graneleiro. O problema representa vários aspectos de um terminal portuário: desde a chegada dos produtos no terminal até a sequência de atendimento de navios, e as mais importantes operações realizadas.

Como contribuição, esta dissertação propôs dois modelos integrados de planejamento, sequenciamento, alocação de pátio e alocação de berço, bem como uma abordagem alternativa de solução para cada uma das formulações. Ambos modelos matemáticos são baseados e estendem o apresentado no trabalho [Menezes, Mateus e Ravetti \(2017\)](#).

As formulações matemáticas propostas são ricas em detalhes realistas, mas também é flexível, podendo ser adaptada para diversos terminais portuários de graneis sólidos (carvão, minério de ferro e grãos).

A abordagem de solução para a formulação matemática PIPSAPAB-I foi baseada na decomposição de Dantzig-Wolfe e no método de geração de coluna. Como apenas o método de geração de coluna não foi capaz de gerar uma solução inteira, foi desenvolvida a heurística que combina o método de geração de colunas, uma heurística de mergulho com *backtracking*, uma heurística *relax-and-fix* de arredondamento e algoritmo exato *branch-and-cut*. Os resultados do extenso teste computacional realizado mostram que a heurística apresentou uma performance satisfatória, pois obteve bons limites superiores

para as instâncias de pequeno, médio e grande porte. Para a formulação matemática PIPSAPAB-II foi desenvolvido um algoritmo heurístico (CMH) que combina heurística de mergulho com *backtracking*, heurísticas de *relax-and-fix*, *local branching* e heurística *rolling horizon*. Esta abordagem também produziu bons resultados, no entanto para instâncias com porte relativamente menor em relação ao testes computacionais da heurística baseada na geração de coluna.

Para validar e testar a desempenho do modelo e os algoritmos, foram conduzidos diversos experimentos computacionais com diversas instâncias do problema variando oferta, demanda, número de rotas, quantidades de produtos, número de períodos, quantidade de navios na fila de atendimento, comprimento de navio e número de berços. Como esperado, solucionar ambas formulações matemáticas via *solver* CPLEX é totalmente inviável, logo justificando o desenvolvimento de abordagens alternativas de solução. Vale destacar que para a formulação PIPSAPAB-I, o *solver* encontrou dificuldades para resolver até mesmo as instâncias de menor porte.

As formulações PIPSAPAB-I e PIPSAPAB-II foram projetadas para modelar o mesmo problema integrado. Portanto, possuem consideráveis semelhanças, porém os modelos matemáticos apresentam diferenças e a principal são as restrições de sequenciamento. A formulação PIPSAPAB-II considera um sequenciamento com microperíodos fixos e alocados de maneira consecutiva para uma determinada rota ativa, enquanto que a formulação PIPSAPAB-I apresenta um sequenciamento com microperíodos variáveis e alocados de forma não consecutiva para uma determinada rota ativa. As considerações do sequenciamento da formulação PIPSAPAB-II, tornam esta mais próximo a da realidade de um terminal portuário, porém a abordagem utilizada no sequenciamento proposto na formulação PIPSAPAB-I também é uma boa aproximação com a realidade. Do ponto de vista dos testes computacionais, verificou-se que o sequenciamento considerado pela formulação PIPSAPAB-II reduz a capacidade de transporte de produtos pelas rotas.

7.2 Trabalhos futuros

Os trabalhos futuros podem se concentrar em melhorias nos algoritmos propostos e novas formulações. Para o algoritmo CGBH, as pesquisas podem estar relacionadas à elaboração de restrições redundantes capazes de ajustar a formulação, bem como à utilização de planos de cortes, sendo ambas estratégias utilizadas para produzir limites mais fortes que os obtidos até o momento. Durante o desenvolvimento deste trabalho, foi elaborada uma decomposição da formulação PIPSAPAB-II, porém a estratégias criadas não apresentaram bons resultados. Com isso, pesquisas futuras poderiam investir na decomposição da atual formulação, ou mesmo, propor uma nova formulação e decomposição. Com relação às novas formulações, alterações poderiam ser feitas na formulação PIPSAPAB-II para aplicar

algum método de decomposição. Outras frentes de pesquisa são considerar incerteza em relação a oferta e demanda, falha e tempo *setup* nos equipamentos.

Referências

AGRA, A.; OLIVEIRA, M. Mip approaches for the integrated berth allocation and quay crane assignment and scheduling problem. **European Journal of Operational Research**, v. 264, 05 2017. Citado 3 vezes nas páginas 6, 7 e 45.

ALFIERI, A. et al. Column generation for minimizing total completion time on a single machine with parallel batching. **IFAC-PapersOnLine**, v. 52, n. 13, p. 969 – 974, 2019. ISSN 2405-8963. 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2405896319312959>>. Citado na página 26.

ANGELELLI, E. et al. A reclaimer scheduling problem arising in coal stockyard management. **Journal of Scheduling**, Springer US, v. 19, 10 2016. Citado na página 9.

ANTAQ. **Relatório de Gestão 2019**. [s.n.], 2019. 96 p. Disponível em: <<http://portal.antaq.gov.br/wp-content/uploads/2020/02/Relatório-de-Gest~ao-2019-site.pdf>>. Citado na página 10.

BARBOSA, F. **Avanços recentes ao problema de alocação de berços**. Tese (Doutorado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, SP, 2018. Disponível em: <<http://www.repositorio.unicamp.br/handle/REPOSIP/331235>>. Citado na página 5.

BARROS, V. H. et al. Model and heuristic for berth allocation in tidal bulk ports with stock level constraints. **Computers Industrial Engineering**, v. 60, n. 4, p. 606 – 613, 2011. ISSN 0360-8352. Citado 3 vezes nas páginas 7, 17 e 21.

BICHOU, K. **Port Operations, Planning and Logistics**. [S.l.]: Informa Law from Routledge, 2013. (Lloyd's Practical Shipping Guides). ISBN 184311805X,9781843118053. Citado na página 1.

BIERWIRTH, C.; MEISEL, F. A survey of berth allocation and quay crane scheduling problems in container terminals. **European Journal of Operational Research**, v. 202, p. 615–627, 05 2010. Citado na página 5.

BIERWIRTH, C.; MEISEL, F. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. **European Journal of Operational Research**, v. 244, 08 2015. Citado na página 5.

BOLAND, N.; GULCZYNSKI, D.; SAVELSBERGH, M. A stockyard planning problem. **EURO Journal on Transportation and Logistics**, v. 1, 09 2012. Citado na página 8.

BURDETT, R. et al. A flexible job shop scheduling approach with operators for coal export terminals – a mature approach. **Computers Operations Research**, v. 115, p. 104834, 11 2019. Citado na página 9.

BUTIKOV, E. A dynamical picture of the oceanic tides. **American Journal of Physics - AMER J PHYS**, v. 70, p. 1001–1011, 10 2002. Citado na página 13.

CARLO, H. J.; VIS, I. F.; ROODBERGEN, K. J. Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. **European Journal of Operational Research**, v. 236, n. 1, p. 1 – 13, 2014. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221713009405>>. Citado na página 5.

CASAZZA, M.; CESELLI, A. Column generation for the variable cost and size bin packing problem with fragmentation. **Electronic Notes in Discrete Mathematics**, v. 55, p. 61 – 64, 2016. ISSN 1571-0653. 14th Cologne-Twente Workshop on Graphs and Combinatorial Optimization (CTW16). Disponível em: <<http://www.sciencedirect.com/science/article/pii/S157106531630172X>>. Citado na página 26.

CHRISTENSEN, J.; ERERA, A.; PACINO, D. A rolling horizon heuristic for the stochastic cargo mix problem. **Transportation Research Part E: Logistics and Transportation Review**, v. 123, p. 200–220, 2019. ISSN 1366-5545. Citado na página 45.

DANTZIG, G. B.; WOLFE, P. Decomposition principle for linear programs. **Operations Research**, v. 8, n. 1, p. 101–111, 1960. Disponível em: <<https://doi.org/10.1287/opre.8.1.101>>. Citado na página 25.

DARWICHE, M. et al. A local branching heuristic for solving a graph edit distance problem. **Computers Operations Research**, v. 106, p. 225–235, 2019. ISSN 0305-0548. Citado na página 42.

DE, A. et al. A hybrid dynamic berth allocation planning problem with fuel costs considerations for container terminal port using chemical reaction optimization approach. **Annals of Operations Research**, v. 290, 07 2020. Citado na página 7.

ERNST, A. et al. Mathematical models for the berth allocation problem in dry bulk terminals. **Journal of Scheduling**, 02 2017. Citado na página 7.

FAIZ, T. I.; VOGIATZIS, C.; NOOR-E-ALAM, M. A column generation algorithm for vehicle scheduling and routing problems. **Computers Industrial Engineering**, v. 130, p. 222 – 236, 2019. ISSN 0360-8352. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0360835219301172>>. Citado na página 26.

FISCHETTI, M.; LODI, A. Local branching. **Mathematical Programming**, v. 98, p. 23–47, 09 2003. Citado 3 vezes nas páginas 41, 42 e 43.

FRANSOO, J.; LEE, C. The critical role of ocean container transport in global supply chain performance. **Production and Operations Management**, v. 22, 03 2013. Citado na página 5.

FREY, M.; KOLISCH, R.; ARTIGUES, C. Column generation for outbound baggage handling at airports. **Transportation Science**, v. 51, n. 4, p. 1226–1241, 2017. Disponível em: <<https://doi.org/10.1287/trsc.2017.0739>>. Citado na página 26.

- GHAREHGOZLI, A.; ROY, D.; KOSTER, R. D. Sea container terminals: New technologies and or models. **Maritime Economics Logistics**, v. 18, 04 2015. Citado na página 5.
- GSCHWIND, T. et al. Bidirectional labeling in column-generation algorithms for pickup-and-delivery problems. **European Journal of Operational Research**, v. 266, n. 2, p. 521 – 530, 2018. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221717308780>>. Citado na página 26.
- GUAN, Y.; YANG, K.-h.; ZHOU, Z. The crane scheduling problem: Models and solution approaches. **Annals of Operations Research - Annals OR**, v. 203, p. 1–21, 03 2013. Citado na página 5.
- HAN, C. hwan. Assessing the impacts of port supply chain integration on port performance. **The Asian Journal of Shipping and Logistics**, v. 34, n. 2, p. 129 – 135, 2018. ISSN 2092-5212. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2092521218300270>>. Citado na página 1.
- HARVEY, W. D.; GINSBERG, M. L. Limited discrepancy search. In: **IJCAI**. [S.l.: s.n.], 1995. Citado na página 31.
- HERNANDEZ, F. et al. A local branching matheuristic for the multi-vehicle routing problem with stochastic demands. **Journal of Heuristics**, v. 25, n. 2, p. 215–245, April 2019. Citado na página 42.
- HIMMICH, I.; HALLAOUI, I. E.; SOUMIS, F. Primal column generation framework for vehicle and crew scheduling problems. **Networks**, v. 75, n. 3, p. 291–309, 2020. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/net.21925>>. Citado na página 26.
- HU, D.; YAO, Z. Stacker-reclaimer scheduling in a dry bulk terminal. **International Journal of Computer Integrated Manufacturing**, Taylor Francis, v. 25, n. 11, p. 1047–1058, 2012. Disponível em: <<https://doi.org/10.1080/0951192X.2012.684707>>. Citado na página 8.
- IBRAHIM, A. et al. Solving some variants of vehicle routing problem with branch-and-cut and column generation algorithms. **Open Journal of Mathematical Sciences**, v. 4, 03 2020. Citado na página 26.
- JANACEK, J. et al. Optimization of periodic crew schedules with application of column generation method. **Transportation Research Part C: Emerging Technologies**, v. 83, p. 165 – 178, 2017. ISSN 0968-090X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0968090X17301894>>. Citado na página 26.
- KASM, O. A.; DIABAT, A.; CHENG, T. C. E. The integrated berth allocation, quay crane assignment and scheduling problem: mathematical formulations and a case study. **Annals of Operations Research**, 01 2019. Citado na página 7.
- Koháni, M.; Janáček, J. Acceleration strategies of the column generation method for the crew scheduling problem. In: **2017 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)**. [S.l.: s.n.], 2017. p. 54–57. Citado na página 26.
- KRIMI, I. et al. A rolling horizon approach for the integrated multi-quays berth allocation and crane assignment problem for bulk ports. **International Journal of Industrial Engineering Computations**, p. 577–591, 01 2019. Citado na página 7.

KULKARNI, S. et al. A new formulation and a column generation-based heuristic for the multiple depot vehicle scheduling problem. **Transportation Research Part B: Methodological**, v. 118, p. 457 – 487, 2018. ISSN 0191-2615. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0191261518301115>>. Citado na página 26.

LIANG, Z. et al. A column generation-based heuristic for aircraft recovery problem with airport capacity constraints and maintenance flexibility. **Transportation Research Part B: Methodological**, v. 113, p. 70 – 90, 2018. ISSN 0191-2615. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0191261517310421>>. Citado na página 26.

LIGTERINGEN, H.; VELSINK, H. **Ports and Terminals**. VSSD, 2012. ISBN 9789065623041. Disponível em: <<https://books.google.com.br/books?id=G0OLngEACAAJ>>. Citado na página 4.

MAHVASH, B.; AWASTHI, A.; CHAUHAN, S. A column generation-based heuristic for the three-dimensional bin packing problem with rotation. **Journal of the Operational Research Society**, Taylor Francis, v. 69, n. 1, p. 78–90, 2018. Disponível em: <<https://doi.org/10.1057/s41274-017-0186-7>>. Citado na página 26.

MENEZES, G. C.; MATEUS, G. R.; RAVETTI, M. G. A hierarchical approach to solve a production planning and scheduling problem in bulk cargo terminal. **Computers Industrial Engineering**, v. 97, p. 1 – 14, 2016. ISSN 0360-8352. Citado na página 8.

MENEZES, G. C.; MATEUS, G. R.; RAVETTI, M. G. A branch and price algorithm to solve the integrated production planning and scheduling in bulk ports. **European Journal of Operational Research**, v. 258, n. 3, p. 926 – 937, 2017. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221716307123>>. Citado 5 vezes nas páginas 3, 8, 13, 26 e 62.

MUNARI, P.; SAVELSBERGH, M. A Column Generation-Based Heuristic for the Split Delivery Vehicle Routing Problem with Time Windows. **SN Operations Research Forum**, v. 1, n. 4, p. 1–24, October 2020. Disponível em: <https://ideas.repec.org/a/spr/snopef/v1y2020i4d10.1007_s43069-020-00026-z.html>. Citado na página 26.

MURTY, K. G. et al. A decision support system for operations in a container terminal. **Decision Support Systems**, v. 39, n. 3, p. 309 – 332, 2005. ISSN 0167-9236. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167923603001465>>. Citado na página 5.

NGUYEN, S. et al. A hybrid differential evolution algorithm with column generation for resource constrained job scheduling. **Computers Operations Research**, v. 109, p. 273 – 287, 2019. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054819301224>>. Citado na página 26.

OZTURK, O. A truncated column generation algorithm for the parallel batch scheduling problem to minimize total flow time. **European Journal of Operational Research**, v. 286, n. 2, p. 432 – 443, 2020. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221720302587>>. Citado na página 26.

PAPADIMITRIOU, D.; FORTZ, B. A rolling horizon heuristic for the multiperiod network design and routing problem. **Networks**, v. 66, n. 4, p. 364–379, 2015. Citado na página 45.

- PEI, Z. et al. A column generation-based approach for proportionate flexible two-stage no-wait job shop scheduling. **International Journal of Production Research**, Taylor Francis, v. 58, n. 2, p. 487–508, 2020. Disponível em: <<https://doi.org/10.1080/00207543.2019.1597291>>. Citado na página 26.
- RESTREPO, M. I.; GENDRON, B.; ROUSSEAU, L.-M. Combining benders decomposition and column generation for multi-activity tour scheduling. **Computers Operations Research**, v. 93, p. 151 – 165, 2018. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054818300145>>. Citado na página 26.
- ROBENEK, T. et al. A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports. **European Journal of Operational Research**, v. 235, n. 2, p. 399 – 411, 2014. ISSN 0377-2217. Maritime Logistics. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221713006802>>. Citado 2 vezes nas páginas 7 e 8.
- Rocha de Paula, M. et al. Throughput optimisation in a coal export system with multiple terminals and shared resources. **Computers Industrial Engineering**, v. 134, p. 37 – 51, 2019. ISSN 0360-8352. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0360835219302888>>. Citado na página 8.
- SADYKOV, R. et al. Primal heuristics for branch and price: The assets of diving methods. **INFORMS Journal on Computing**, v. 31, n. 2, p. 251–267, 2019. Disponível em: <<https://doi.org/10.1287/ijoc.2018.0822>>. Citado na página 31.
- SAID, G. A. E.-N.; EL-HORBARTY, E.-S. An optimization methodology for container handling using genetic algorithm. **Procedia Computer Science**, v. 65, p. 662–671, 12 2015. Citado na página 5.
- SAMAVATI, M. et al. A local branching heuristic for the open pit mine production scheduling problem. **European Journal of Operational Research**, v. 257, n. 1, p. 261–271, 2017. ISSN 0377-2217. Citado na página 42.
- Schmidt, T. et al. An application of the local branching to the identical parallel machines scheduling problem. **IEEE Latin America Transactions**, v. 17, n. 06, p. 1047–1054, 2019. Citado na página 42.
- SHANG, X. T.; CAO, J. X.; REN, J. A robust optimization approach to the integrated berth allocation and quay crane assignment problem. **Transportation Research Part E: Logistics and Transportation Review**, v. 94, p. 44 – 65, 2016. ISSN 1366-5545. Citado 2 vezes nas páginas 6 e 7.
- STAHLBOCK, R.; VOSS, S. Operations research at container terminals: A literature update. **Operations Research-Spektrum**, v. 30, p. 1–52, 04 2008. Citado na página 5.
- STEENKEN, D.; VOSS, S.; STAHLBOCK, R. Container terminal operation and operations research - a classification and literature review. **OR Spectrum**, v. 26, p. 3–49, 01 2004. Citado na página 5.
- TANG, L.; SUN, D.; LIU, J. Integrated storage space allocation and ship scheduling problem in bulk cargo terminals. **IIE Transactions**, Taylor Francis, v. 48, n. 5, p. 428–439, 2016. Disponível em: <<https://doi.org/10.1080/0740817X.2015.1063791>>. Citado 2 vezes nas páginas 7 e 8.

TAŞ, D. Electric vehicle routing with flexible time windows: a column generation solution approach. **Transportation Letters**, Taylor Francis, v. 0, n. 0, p. 1–7, 2020. Disponível em: <<https://doi.org/10.1080/19427867.2020.1711581>>. Citado na página 26.

TCHOUPPO, M. N. et al. An efficient column-generation algorithm for a new fleet size and mix pickup and delivery problem with time windows. **IFAC-PapersOnLine**, v. 51, n. 9, p. 440 – 445, 2018. ISSN 2405-8963. 15th IFAC Symposium on Control in Transportation Systems CTS 2018. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S240589631830795X>>. Citado na página 26.

UMANG, N.; BIERLAIRE, M.; VACCA, I. Exact and heuristic methods to solve the berth allocation problem in bulk ports. **Transportation Research Part E: Logistics and Transportation Review**, v. 54, 07 2013. Citado na página 7.

UNCTAD. **Review of Maritime Transport 2020**. [s.n.], 2020. Disponível em: <https://unctad.org/system/files/official-document/rmt2020_en.pdf>. Citado na página 1.

UNSAI, O. Reclaimer scheduling in dry bulk terminals. **IEEE Access**, PP, p. 1–1, 05 2020. Citado na página 9.

UNSAI, O.; OĞUZ, C. An exact algorithm for integrated planning of operations in dry bulk terminals. **Transportation Research Part E: Logistics and Transportation Review**, v. 126, p. 103–121, 06 2019. Citado 4 vezes nas páginas 2, 4, 7 e 17.

VACCA, I.; SALANI, M.; BIERLAIRE, M. An exact algorithm for the integrated planning of berth allocation and quay crane assignment. **Transportation Science**, v. 47, p. 148–161, 05 2013. Citado na página 6.

VIEIRA, J. H. **Modernização da gestão portuária e planejamento operacional integrado por meio de indicadores de desempenho**. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, Escola Politécnica, 2018. Citado na página 10.

VIS, I. F.; de Koster, R. Transshipment of containers at a container terminal: An overview. **European Journal of Operational Research**, v. 147, n. 1, p. 1 – 16, 2003. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S037722170200293X>>. Citado na página 5.

WANG, C.; LU, X.-w.; SITTERS, R. Scheduling reclaimer operations in the stockyard to minimize makespan. **Acta Mathematicae Applicatae Sinica, English Series**, v. 34, p. 597–609, 07 2018. Citado na página 9.

WANG, K. et al. Column generation for the integrated berth allocation, quay crane assignment, and yard assignment problem. **Transportation Science**, v. 52, n. 4, p. 812–834, 2018. Disponível em: <<https://doi.org/10.1287/trsc.2018.0822>>. Citado 3 vezes nas páginas 6, 7 e 26.

WOLSEY, L. **Integer Programming**. [S.l.]: Wiley, 1998. (Wiley Series in Discrete Mathematics and Optimization). ISBN 9780471283669. Citado na página 33.

YANG, Z. et al. An effective hybrid approach to the two-stage capacitated facility location problem. **European Journal of Operational Research**, v. 275, n. 2, p. 467–480, 2019. ISSN 0377-2217. Citado na página 42.

ZHANG, J.; DRIDI, M.; El Moudni, A. Column-generation-based heuristic approaches to stochastic surgery scheduling with downstream capacity constraints. **International Journal of Production Economics**, v. 229, p. 107764, 2020. ISSN 0925-5273. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925527320301468>>. Citado na página 26.

Zhang, Q. et al. A column generation-based algorithm for two-stage, two-dimensional bin packing problem with a variant variable sized constraint. In: **2017 36th Chinese Control Conference (CCC)**. [S.l.: s.n.], 2017. p. 2841–2845. ISSN 1934-1768. Citado na página 26.

ZHEN, L. et al. Column generation for low carbon berth allocation under uncertainty. **Journal of the Operational Research Society**, Taylor Francis, v. 0, n. 0, p. 1–16, 2020. Disponível em: <<https://doi.org/10.1080/01605682.2020.1776168>>. Citado na página 26.