



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Catarina Junqueira

Modelagem Integrada do Plano de Estiva para Múltiplos Portos e Realocação de Contêineres em Pátios Portuários

Campinas

2019



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Catarina Junqueira

Modelagem Integrada do Plano de Estiva para Múltiplos Portos e Realocação de Contêineres em Pátios Portuários

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutora em Engenharia Elétrica, na Área de Automação.

Orientador: Prof. Dr. Takaaki Ohishi

Co-orientador Prof. Dr. Anibal Tavares de Azevedo

Este exemplar corresponde à versão final da tese defendida pelo aluno Catarina Junqueira, e orientada pelo Prof. Dr. Takaaki Ohishi

Campinas

2019

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

J968m Junqueira, Catarina, 1990-
Modelagem integrada do plano de estiva para múltiplos portos e realocação de contêineres em pátios portuários / Catarina Junqueira. – Campinas, SP : [s.n.], 2019.

Orientador: Takaaki Ohishi.

Coorientador: Anibal Tavares de Azevedo.

Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Navios porta-containers. 2. Transporte por containers. 3. Áreas portuárias. 4. Portos. 5. Otimização. I. Ohishi, Takaaki, 1955-. II. Tavares de Azevedo, Anibal, 1977-. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Integrated modeling of the multi-port stowage planning and container relocation in port yards

Palavras-chave em inglês:

Container ship

Transport by containers

Port areas

Ports

Optimization

Área de concentração: Automação

Titulação: Doutora em Engenharia Elétrica

Banca examinadora:

Takaaki Ohishi [Orientador]

Cléber Damião Rocco

Cristiano Torezzan

Luiz Leduíno de Salles Neto

Antônio Augusto Chaves

Data de defesa: 23-08-2019

Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0001-6724-8199>

- Currículo Lattes do autor: <http://lattes.cnpq.br/1050658200221972>

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidata: Catarina Junqueira **RA:** 093566

Data da Defesa: 23 de Agosto de 2019

Título da Tese: “Modelagem Integrada do Plano de Estiva para Múltiplos Portos e Realocação de Contêineres em Pátios Portuários”

Prof. Dr. Takaaki Ohishi

Prof. Dr. Cléber Damião Rocco

Prof. Dr. Cristiano Torezzan

Prof. Dr. Luiz Leduíno de Salles Neto

Prof. Dr. Antônio Augusto Chaves

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de PósGraduação da Faculdade de Engenharia Elétrica e de Computação.

Dedico esta tese às mulheres na ciência.

Agradecimentos

Agradeço à minha família, a quem devo grande parte das minhas conquistas, por ter me dado condições de me dedicar exclusivamente aos estudos e à pesquisa.

Ao meu namorado, Miguel, pelo companheirismo e apoio durante esta jornada.

Aos professores Aníbal e Takaaki, pela orientação e contribuição na minha formação acadêmica.

Aos meus amigos do COSE, pelos bons momentos que passamos juntos no laboratório.

Ao professor Cléber, pelas dicas e contribuições que foram fundamentais no final deste trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

Nesta tese é desenvolvido um modelo de otimização que integra o problema do plano de estiva para múltiplos portos com o problema de realocação de contêineres no pátio portuário, chamado de CRP-MPSP. O CRP-MPSP é formulado como um modelo puramente binário, que deverá encontrar uma sequência de movimentação dos contêineres de forma que o número de remanejamentos, considerando toda a jornada de um navio e os pátios portuários associados, seja minimizado. O uso de variáveis binárias para representar o estado do navio e dos pátios torna este um problema de alta complexidade. A modelagem aqui apresentada ainda não foi desenvolvida na literatura como um modelo unificado, visto que tais problemas tendem a ser tratados de forma particionada. Duas regras heurísticas são propostas para gerar soluções factíveis que serão utilizadas como warm-start pelo método de solução exato. O CRP-MPSP é implementado em dois solvers comerciais diferentes e os resultados para instâncias geradas aleatoriamente são apresentados. Extensivos testes computacionais são realizados. Os resultados indicam que a abordagem de solução pode obter soluções ótimas em instâncias de tamanho pequeno e soluções de boa qualidade em instâncias de tamanho maior, em tempo computacional razoável.

Palavras-chaves: Navios porta-contêineres; Transporte por contêineres; Áreas portuárias; Portos; Otimização

Abstract

This thesis develops an optimization model that integrates the multi-port stowage planning problem with the container relocation problem at the port yard, called CRP-MPSP. The CRP-MPSP is formulated as a purely binary model, that shall find the containers move sequence so that the number of relocations, considering the entire journey of a ship and the associated port yards, is minimized. The use of binary variables to represent the state of the ship and the yards make this a highly complex problem. The model presented here has not yet been developed in the literature as a integrated model, given that such problems tend to be treated in a partitioned way. Two heuristic rules are proposed to quickly generate feasible solutions to be used as warm-start for the exact solution technique. The CRP-MPSP is implemented in two different commercial solvers and the results for randomly generated instances are presented. Extensive computational tests are performed. The results indicate that the solution approach can obtain optimal solutions in small size instances, and good quality solutions in larger size instances within reasonable computation time.

Keywords: Container ship; Transport by containers; Port areas; Ports; Optimization

Lista de ilustrações

Figura 3.1 – Principais áreas de um terminal de contêineres. Fonte: Adaptado de (CARLO <i>et al.</i> , 2014)	16
Figura 3.2 – Fluxos de contêineres entre o navio e o pátio portuário.	17
Figura 3.3 – Representação de um bloco em um pátio de contêineres. Adaptada de (WAN <i>et al.</i> , 2009).	19
Figura 3.4 – Vista em corte transversal de uma baía. Fonte: Adaptado de (JIN <i>et al.</i> , 2015)	19
Figura 3.5 – A suposição do CRP restrito. Adaptado de (TRICOIRE <i>et al.</i> , 2018). .	20
Figura 3.6 – A suposição do CRP irrestrito. Adaptado de (TRICOIRE <i>et al.</i> , 2018). .	21
Figura 3.7 – Estrutura do local de armazenamento de um navio-contêiner. Adaptado de (BIERWIRTH; MEISEL, 2010).	27
Figura 3.8 – Vista em corte transversal de uma baía. Adaptado de (BIERWIRTH; MEISEL, 2010).	27
Figura 3.9 – Fluxo de contêineres entre as partes do navio e do pátio e suas operações correspondentes.	35
Figura 3.10–Primeiras operações após a chegada do navio no porto 2: descarregamento.	36
Figura 3.11–Carregamento no navio dos contêineres do pátio de exportação do porto 2.	37
Figura 3.12–Política alternativa à apresentada na Figura 3.10.	38
Figura 3.13–Carregamento no navio dos contêineres do pátio temporário e de exportação do porto 2.	38
Figura 4.1 – Exemplo de uma rota de um navio por 5 portos.	40
Figura 4.2 – Exemplo do descarregamento de um navio.	41
Figura 4.3 – Exemplo do carregamento de um navio.	41
Figura 5.1 – Pátio do primeiro porto da instância IT_A	52
Figura 5.2 – Pátio do primeiro porto da instância IT_B	52
Figura 5.3 – Configuração do Navio em cada porto p da rota do navio na instância IT_A	54
Figura 5.4 – Configuração do Navio em cada porto p da rota do navio na instância IT	55
Figura 5.5 – Sequência de retirada dos contêineres do pátio do primeiro porto da instância IT_B , dado o plano de estiva apresentado na Figura 5.4 - Parte 1.	57

Figura 5.6 – Sequência de retirada dos contêineres do pátio do primeiro porto da instância IT_B , dado o plano de estiva apresentado na Figura 5.4 - Parte 2.	57
Figura 5.7 – Sequência de retirada dos contêineres do pátio do primeiro porto da instância IT_B , dado o plano de estiva apresentado na Figura 5.4 - Parte 3.	58
Figura 6.1 – Primeiros passos da Regra Heurística HR_1	61
Figura 6.2 – Regra Heurística HR_1 : Alocando o contêiner \hat{n} na coluna c^* do navio.	62
Figura 6.3 – Sequência de retirada dos contêineres de um pátio portuário.	65
Figura 6.4 – Representação da ocupação de navio em cada porto p de uma rota com três portos.	65
Figura 6.5 – Representação da retirada de contêineres do pátio e carregamento no navio.	67
Figura 7.1 – Parte do relatório de progresso do Cplex para a instância $7B$	72
Figura 7.2 – Parte do relatório de progresso do Gurobi para a instância $7B$	73
Figura 7.3 – Parte do relatório de progresso do Gurobi para a instância $4A$	73
Figura 7.4 – Parte do relatório de progresso do Cplex para a instância $4A$	74
Figura 7.5 – Comparativo entre tempo de resolução e o número de variáveis e restrições das instâncias $1A$ a $8A$ obtido pelo Cplex.	75
Figura 7.6 – Comparativo entre tempo de resolução e o número de variáveis e restrições das instâncias $1B$ a $8B$ obtido pelo Cplex.	75
Figura 7.7 – Cplex x Gurobi: Comparativo dos tempos de resolução.	76

Lista de tabelas

Tabela 5.1 – Instâncias de Teste	50
Tabela 5.2 – Resultados das instâncias IT_A e IT_B	53
Tabela 5.3 – Resultados da instância IT	55
Tabela 7.1 – Instâncias de teste do grupo A	70
Tabela 7.2 – Instâncias de teste do grupo B	70
Tabela 7.3 – Resultados Computacionais do CRP-MPSP: Cplex x Gurobi	71
Tabela 7.4 – Resultados Computacionais do CRP-MPSP: Cplex x Gurobi	71
Tabela 7.5 – Resultados das regras heurísticas HR_1 e HR_2	77
Tabela 7.6 – Instâncias de teste de larga escala.	77
Tabela 7.7 – Resultados computacionais das regras heurísticas em instâncias de larga escala.	78
Tabela 7.8 – Resultados computacionais do CRP-MPSP com <i>warm-start</i> : Cplex x Gurobi	79
Tabela 7.9 – Resultados computacionais do CRP-MPSP com <i>warm-start</i> : Cplex x Gurobi	79
Tabela 8.1 – Comparação do tamanho dos modelos.	84
Tabela 8.2 – Comparação do tamanho dos modelos.	84
Tabela 8.3 – Resultados computacionais do CRP-MPSP-II com <i>warm-start</i> : Cplex x Gurobi.	85
Tabela 8.4 – Resultados computacionais do CRP-MPSP-II com <i>warm-start</i> : Cplex x Gurobi.	85

Sumário

1	Introdução	13
2	Objetivos	15
2.1	Objetivo geral	15
2.2	Objetivos específicos	15
3	Descrição dos Problemas	16
3.1	Fluxo de Contêineres	16
3.2	O Problema de Realocação de Contêineres	18
3.3	O Problema do Plano de Estiva para Múltiplos Portos	26
3.4	Integração	32
4	Modelagem do CRP-MPSP	40
4.1	Modelo matemático do CRP-MPSP	41
4.1.1	Variáveis do Problema	44
4.1.2	Função Objetivo	45
4.1.3	Restrições	45
5	Justificativa	50
6	Técnica de Solução	59
6.1	Regras Heurísticas	59
6.1.1	Regra Heurística HR_1	60
6.1.2	Regra Heurística HR_2	64
6.2	<i>Warm-Start</i>	64
7	Resultados Computacionais	69
7.1	Resultados do CRP-MPSP	69
7.2	Resultados das Regras Heurísticas HR_1 e HR_2	76
7.3	Resultados do Warm-Start	78
8	Uma Formulação Alternativa para o CRP-MPSP	81
8.1	Resultados Computacionais do CRP-MPSP-II	83
9	Conclusão	87
	Conclusão	87
	Referências	89

1 Introdução

O transporte marítimo é o meio de transporte mais importante para o comércio internacional, cujo crescimento contínuo representa maior pressão nos terminais portuários.

Uma pesquisa feita por (BERNHOFEN *et al.*, 2013) apontou que o comércio em contêineres tem sido mais importante para a globalização do que os acordos bilaterais de livre comércio, pois explica um aumento de 320% no comércio bilateral nos países industrializados passados cinco anos de sua adoção, e um aumento de 790% passados 20 anos. De acordo (ZHEN *et al.*, 2013), desde os anos 90, o tráfego mundial de contêineres tem crescido três vezes mais do que o PIB mundial.

Segundo dados dos relatórios da Conferência das Nações Unidas sobre Comércio e Desenvolvimento (UNCTAD), estima-se que os volumes mundiais de comércio marítimo tenham ultrapassado os 10 bilhões de toneladas em 2015, dos quais aproximadamente 16% são de mercadorias conteineirizadas.

No Brasil, no ano de 2016 foram movimentados 9.289.501 TEU (UNCTADstat, 2018a), representando 1,46% da movimentação mundial (UNCTADstat, 2018b). TEU significa Unidades Equivalentes de Vinte Pés (*Twenty-Foot Equivalent Units*). Um navio com capacidade de 8000 TEU pode carregar 8000 contêineres de vinte pés (AZEVEDO *et al.*, 2011). Os contêineres de quarenta pés são medidos como 2 TEU.

Aliado à retomada do crescimento no comércio internacional, o tamanho dos navios-contêineres também tem crescido continuamente, a fim de aumentar os benefícios da economia de escala e manter os preços de frete competitivos. Em maio de 2017 foi entregue o maior navio-contêiner do mundo, o *OOCL Hong Kong*, que possui capacidade de 21.413 TEU.

Para (UNCTAD/RMT, 2017), haverá provavelmente a necessidade de portos e companhias marítimas cooperarem entre si, uma vez que o aumento no tamanho dos navios pode trazer também picos mais altos no tráfego de contêineres, maior pressão sobre as operações de movimentação dos contêineres, aumento dos custos operacionais, e outros transtornos.

Nestas circunstâncias, melhorar a produtividade no porto, a fim de reduzir custos e alcançar uma melhor integração no comércio global, se torna crucial para manter a competitividade econômica do setor, uma vez que organização dos terminais de contêineres afeta significativamente toda a cadeia de suprimentos.

Neste contexto, este trabalho trata da integração de dois problemas operacionais deste setor: o problema de realocação de contêineres em pátios portuários (CRP - *Container Relocation Problem*) e o problema do plano de estiva para múltiplos portos (MPSP - *Multi-Port Stowage Planing Problem*). Para isso, é desenvolvido um modelo de otimização inteiro binário, chamado de CRP-MPSP (*Container Relocation Problem with Multi-Port Stowage Planing Problem*).

A originalidade deste trabalho se dá no sentido de que não há na literatura modelo matemático que represente essas atividades de forma integrada, nem uma metodologia que as resolva analiticamente. Mais do que isso, este trabalho tem como principal objetivo mudar uma prática comum e aceita de tratar o CRP e o MPSP de forma particionada ou simplificada. De fato, a maior contribuição é mostrar que o CRP e o MPSP realmente são correlacionados e devem ser otimizados de forma integrada.

Além disso, diferentes regras heurísticas são desenvolvidas para gerar soluções factíveis para o modelo CRP-MPSP de forma rápida. Estas soluções serão utilizadas como *warm-start* no método de solução exato, com o objetivo de acelerar a resolução do CRP-MPSP. Extensivos testes computacionais são realizados, e os resultados obtidos por dois solvers comerciais diferentes, o Cplex e o Gurobi, são comparados e discutidos. Na sequência, são exploradas algumas alternativas que deixam o modelo do CRP-MPSP mais conciso.

O restante do trabalho está organizado da seguinte forma: No Capítulo 2 são apresentados o objetivo geral e os objetivos secundários, em seguida, no Capítulo 3 é feita uma descrição detalhada com revisão da literatura acerca do CRP e do MPSP de forma individual, seguida por uma justificativa da integração de ambos. A modelagem inteira binária do CRP-MPSP é apresentada no Capítulo 4. No Capítulo 5 uma demonstração de que o CRP e o MPSP devem ser resolvidos de forma integrada é fornecida. A abordagem de solução proposta é apresentada no Capítulo 6. No Capítulo 7 são apresentados os resultados dos testes computacionais, e uma formulação alternativa é proposta no Capítulo 8. Finalmente, no Capítulo 9 é feita a conclusão e sugestões de trabalhos futuros.

2 Objetivos

A presente tese tem os seguintes objetivos:

2.1 Objetivo geral

- Desenvolver e resolver um modelo de otimização que integre o MPSP com o CRP.

2.2 Objetivos específicos

Além do objetivo geral, outros objetivos poderão ser atingidos, tais como:

- Apresentar revisão bibliográfica a respeito dos diferentes modelos e métodos com os quais o MPSP e o CRP são abordados e analisados pela literatura;
- Explorar melhorias que deixem o modelo mais conciso;
- Implementar uma abordagem que encontre soluções factíveis de forma rápida para o modelo proposto, e utilizá-las como inicialização para o método de solução exato;
- Apresentar resultados que demonstrem a importância de se considerar de forma integrada o MPSP e CRP.

3 Descrição dos Problemas

Por fins didáticos e um melhor entendimento de cada problema, primeiramente é importante conhecer o fluxo de contêineres em um terminal portuário. Na sequência o CRP e o MPSP serão explicados individualmente. Por fim, a integração entre eles será detalhada e justificada.

3.1 Fluxo de Contêineres

As principais áreas de terminais contêineres são o berço, o cais, a área de transporte, o pátio, e o portão (CARLO *et al.*, 2014), conforme é ilustrado na Figura 3.1.

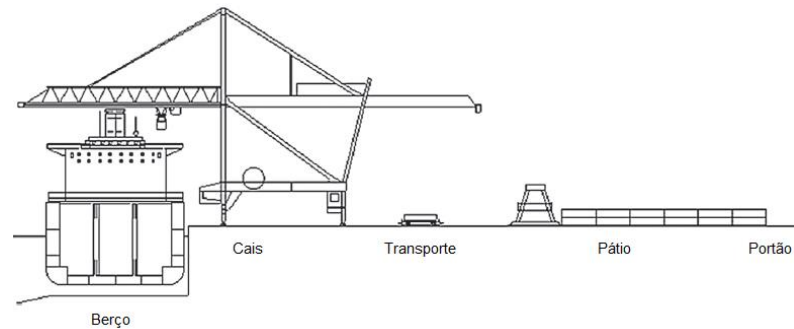


Figura 3.1 – Principais áreas de um terminal de contêineres. Fonte: Adaptado de (CARLO *et al.*, 2014)

Para (CARLO *et al.*, 2014), o berço e as áreas de cais podem ser chamadas de áreas ‘à beira-mar’, as áreas do pátio e dos portões de áreas ‘terrestres’ e a área de transporte é a intersecção das áreas à beira-mar e a terrestre.

Um contêiner pode ser diferenciado pela forma com a qual chega ao terminal de contêineres. Dentre elas, tem-se:

- **Contêineres de Exportação:** Chegam no terminal um a um, através de um caminhão externo ou trem e em uma ordem imprevisível. São armazenados temporariamente em pilhas no pátio e deixam o porto em grandes quantidades a bordo de um navio, em ordem previsível.
- **Contêineres de Importação:** Chegam no terminal a bordo de um navio e seu destino final é o porto onde foram descarregados. Eles são armazenados temporariamente em pilhas em uma área separada do pátio para esperar um caminhão externo ou trem buscá-los. Esses contêineres chegam no porto em grandes quantidades e em uma ordem previsível, mas saem, um a um, em uma ordem imprevisível.

- **Contêineres Temporários:** Chegam no terminal a bordo de um navio e são descarregados apenas para liberar acesso a outros contêineres que foram armazenados embaixo deles na mesma pilha do navio. Estes contêineres são armazenados temporariamente em pilhas no pátio e eventualmente serão novamente carregados no mesmo navio e transportados para seu destino final.
- **Contêineres de Transbordo:** Chegam no terminal a bordo de um navio, entretanto, seu destino final é um porto diferente do qual foram descarregados. Estes contêineres são armazenados temporariamente em pilhas no pátio e eventualmente serão carregados em outros navios e transportados para seus próximos destinos. Tanto a chegada como a saída destes contêineres são relativamente fixas e previsíveis.

O fluxo desses contêineres pelo terminal pode ser visualizado na Figura 3.2.

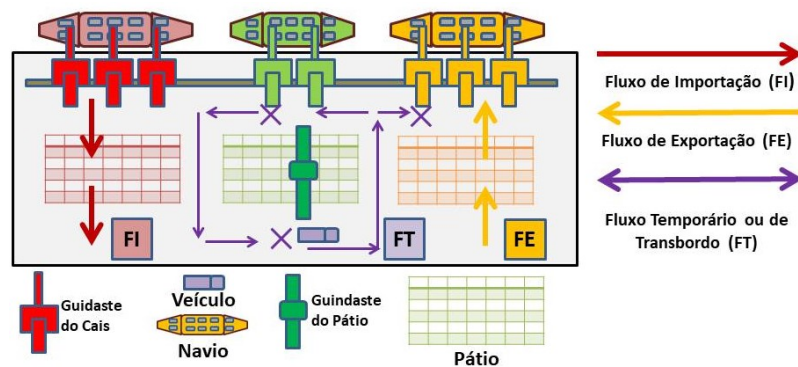


Figura 3.2 – Fluxos de contêineres entre o navio e o pátio portuário.

No pátio, muitas vezes, os contêineres estão separados em áreas de contêineres de exportação, importação, temporários ou de transbordo. Dentro dessas áreas, ainda podem haver sub-divisões, como por exemplo, áreas reservadas para contêineres especiais, como os refrigerados que precisam de uma conexão elétrica, contêineres com cargas perigosas, fora da dimensão ou peso padrão ou ainda vazios ((STEENKEN *et al.*, 2004)). Consequentemente, a forma com a qual cada tipo de contêiner é operado difere de acordo com seu tipo ((CHEN; LU, 2012), (ZHEN *et al.*, 2013)). De um modo geral, a estratégia de armazenamento escolhida visa melhor utilizar as informações relativamente fixas (de entrada ou saída) dos contêineres que estão chegando ao terminal, de modo a armazená-los no local adequado para facilitar as retiradas (ZHEN *et al.*, 2013).

Dessa forma, a Figura 3.2 é útil para entender a necessidade de coordenação entre os diferentes fluxos de contêineres entre um navio e um pátio de contêineres. Para (HA *et al.*, 2019a), operações adequadamente coordenadas podem significar uma maior

produtividade no terminal portuário do que aumentar a capacidade dos equipamentos, por exemplo.

(ZHEN *et al.*, 2013) faz um levantamento sobre as estratégias de armazenamento mais populares para diferentes tipos de contêineres. Por exemplo, segundo os autores, dentro da área dos contêineres de importação, a estratégia de segregação é bastante popular na literatura. Essa estratégia consiste em separar os novos contêineres que estão sendo desembarcados de um certo navio dos contêineres antigos que já estavam no pátio para facilitar a retirada. Enquanto isso, para os contêineres de exportação, uma estratégia popular é a de clusterização. Essa estratégia busca agrupar os contêineres que possuem o horário de retirada próximo para facilitar o processo de carregamento. Em relação aos contêineres de transbordo, a estratégia mais popular é a de agrupar os contêineres com o mesmo navio de destino, chamada de estratégia de consignação.

Neste trabalho, o impacto de diferentes estratégias de armazenamento não será analisado. A premissa adotada é de que os contêineres estão organizados em áreas separadas de contêineres de exportação, importação, temporários ou de transbordo. Em relação aos contêineres de exportação, assume-se que todos os contêineres com o mesmo navio de destino estão agrupados em uma única baía. Mais detalhes sobre a organização dos contêineres adotada serão dados nas seções seguintes.

A seguir, as Subseções 3.2 e 3.3 detalham o CRP e o MPSP.

3.2 O Problema de Realocação de Contêineres

O pátio portuário é o local onde os contêineres ficam estocados temporariamente até que sejam retirados e carregados em um navio, caminhão ou trem. Devido às limitações de espaço, os contêineres são armazenados em pilhas verticais; o que significa que a organização é feita em blocos ou áreas. Cada bloco possui diversas baias que são compostas de um grupo de colunas. A Figura 3.3 detalha um bloco de contêineres em um pátio.

Uma visão transversal de uma baía com suas pilhas verticais é dada na Figura 3.4. Nesta Figura, cada número em uma célula indica a ordem em que o contêiner que a ocupa deve ser removido do pátio, ou seja, o primeiro contêiner a ser removido é o contêiner 1, seguido pelo contêiner 2, e assim por diante até o contêiner 12.

A Figura 3.4 também mostra que a posição $(Linha, Coluna) = (2, 3)$, na qual o contêiner cuja ordem de retirada do pátio é 6, está bloqueando o contêiner alvo que está na posição $(Linha, Coluna) = (1, 3)$ e deve ser retirado primeiro.

Neste caso, pelo menos um movimento de realocação é necessário. Realocações

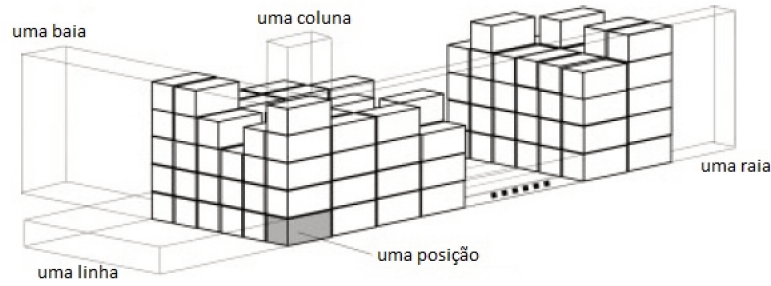


Figura 3.3 – Representação de um bloco em um pátio de contêineres. Adaptada de (WAN *et al.*, 2009).

ou remanejamentos são movimentos que ocorrem durante a retirada de um contêiner que não está no topo de uma pilha de contêineres, e implica que o contêiner de bloqueio seja realocado do topo da pilha na qual se encontra para outra pilha. Dessa forma, o acesso ao contêiner alvo (ou contêiner objetivo) é liberado e o mesmo pode ser retirado. Formalmente, uma realocação é o movimento de um contêiner c de uma pilha de origem s_1 para uma pilha de destino s_2 (JIN *et al.*, 2015).

Bloqueio							Linha
		7					4
		2				12	3
		5	3	6		11	2
		8	4	1	9	10	1
	Coluna	1	2	3	4	5	

Figura 3.4 – Vista em corte transversal de uma baía. Fonte: Adaptado de (JIN *et al.*, 2015)

Realocações são consideradas movimentos improdutivos. Por isso, o número de relocações é um fator importante que afeta a eficiência operacional em terminais de contêineres, e está correlacionado com a sequência de carregamento e a estratégia de manipulação (JI *et al.*, 2015).

Em função disso, o problema de realocação de contêineres no pátio (CRP - *container relocation problem*), também conhecido na literatura como problema de realocação de blocos (BRP - *block relocation problem*), lida com um determinado conjunto de contêineres (ou blocos) homogêneos armazenados em um conjunto de pilhas bidimensionais que são organizadas seguindo a política LIFO (*Last In First Out*), ou seja, o último contêiner que entra na pilha é o primeiro a sair. Dessa forma, o CRP consiste em retirar todos estes contêineres da área de armazenamento em uma ordem pré-definida de forma que o número de remanejamentos seja minimizado.

Em (CASERTA *et al.*, 2012), o BRP é provado ser NP-hard. Isto é feito mostrando que uma decisão em um BRP com linhas e colunas finitas é equivalente a uma

decisão reduzida do problema de Mutual Exclusion Scheduling (MES) em grafos de permutação. Ser NP-hard significa que não são conhecidos algoritmos de tempo polinomial para resolver este problema, então o tempo para encontrar uma solução cresce exponencialmente conforme cresce o tamanho do problema.

Por este motivo, uma ampla gama de heurísticas e meta-heurísticas tem sido aplicadas, como *beam-search* ((WU; TING, 2010), (NISHI; KONISHI, 2010), (TING; WU, 2017), (BACCI *et al.*, 2019)), o método do corredor ((CASERTA; VOSS, 2009), (CASERTA *et al.*, 2011)), algoritmo genético ((HUSSEIN; PETERING, 2012)), colônia de formigas ((JOVANOVIC *et al.*, 2019)), busca em árvore ((FORSTER; BORTFELDT, 2012)) e GRASP((FIRMINO *et al.*, 2019)).

A literatura acerca do CRP contém as seguintes variantes principais: CRP restrito ou CRP irrestrito. No caso restrito, o pressuposto é que os únicos contêineres que podem ser remanejados são aqueles que bloqueiam o próximo contêiner a ser retirado. No caso irrestrito, todos os movimentos factíveis de remanejamento são considerados.

A consideração do CRP restrito é razoável uma vez que permite uma redução do espaço de busca, entretanto pode levar à exclusão de soluções ótimas ((CASERTA *et al.*, 2012)). As Figuras 3.5 e 3.6 esclarecem este ponto.

Repare na Figura 3.5 que, considerando o problema restrito, pelo menos 6 remanejamentos são necessários para se retirar todos os contêineres do pátio, uma vez que para retirar o contêiner 1, os contêineres 3, 4 e 5 foram movidos para cima do contêiner 2, que é o próximo contêiner a ser retirado e por isso terão que ser remanejados novamente. Enquanto isso, considerando o problema irrestrito, como pode ser visto na Figura 3.6, 4 remanejamentos são necessários. Isso acontece porque no caso irrestrito o contêiner 2 pode ser movido para a coluna 1 antes de que sejam feitos os remanejamentos necessários para se retirar o contêiner 1. Com isso, nenhum remanejamento adicional é necessário para que os contêineres remanescentes sejam retirados.

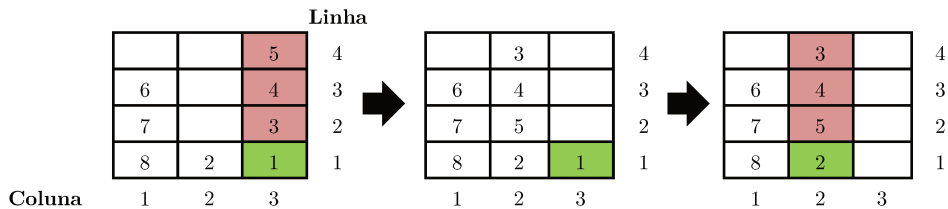


Figura 3.5 – A suposição do CRP restrito. Adaptado de (TRICOIRE *et al.*, 2018).

Neste trabalho, apenas a variante do CRP restrito é considerada.

O modelo de otimização do CRP é introduzido por (KIM; HONG, 2006), e dois métodos são sugeridos para resolvê-lo. O primeiro método é um algoritmo de *branch and*

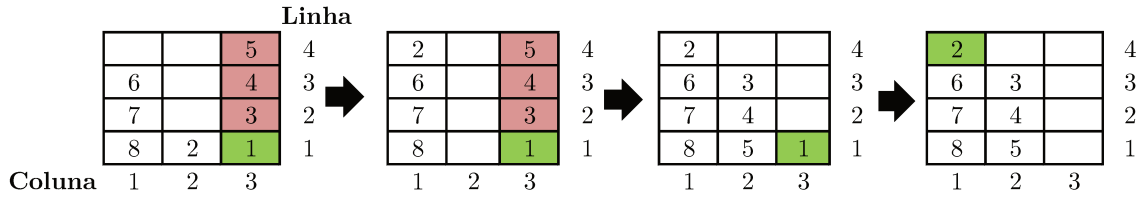


Figura 3.6 – A suposição do CRP irrestrito. Adaptado de (TRICOIRE *et al.*, 2018).

bound e o segundo método é uma regra heurística que usa um estimador para o número esperado de realocações adicionais em cada pilha (ENAR - *expected number of additional relocation*). Para tanto, os autores assumem que: (i) a ordem de retirada dos contêineres é previamente conhecida; (ii) a possibilidade de pré-remanejamentos, para reduzir o número de futuros remanejamentos, é excluída; e (iii) que os blocos somente são remanejados para outras pilhas na mesma área de armazenamento. Os autores apresentam resultados numéricos para instâncias com no máximo seis colunas e cinco linhas e concluem que o número total de realocações calculadas pela regra heurística excede o número encontrado pelo *branch and bound*. No entanto, o tempo computacional utilizado pela regra heurística é muito menor do que o consumido pelo *branch and bound*. Seus resultados são importantes porque serviram como parâmetros de comparação para alguns dos trabalhos que serão apresentados a seguir.

Em (WAN *et al.*, 2009) é apresentado um modelo linear inteiro com variáveis binárias e objetivo de minimizar o número total de remanejamentos. Também é assumido que não há pré-remanejamento dos contêineres e para a retirada de qualquer contêiner, cada contêiner remanejado é movido apenas uma vez. Naturalmente, o modelo necessita de um longo tempo computacional para resolver instâncias grandes. Dessa forma, de modo a obter um melhor *trade-off* entre tempo computacional e qualidade da solução, uma heurística baseada em programação inteira que gera uma solução ao resolver uma série de modelos inteiros reduzidos é proposta. Os autores também fazem uso das versões originais e estendidas de três regras baseadas em índices definidas em (MURTY *et al.*, 2005) e do número esperado de remanejamentos adicionais (ENAR), definido por (KIM; HONG, 2006). As instâncias testadas possuem seis colunas, e entre dois e cinco linhas. Como esperado, o número médio de remanejamentos e o tempo computacional médio aumentam com o tamanho e a densidade de armazenamento. Os autores também aplicaram suas heurísticas em um ambiente dinâmico com chegadas e retiradas de contêineres contínuas. Os resultados computacionais mostraram que a heurística baseada em programação inteira superou todas as heurísticas baseadas em índices que foram propostas pelos autores.

(LEE; LEE, 2010) apresentam uma heurística de três fases para resolver o BRP restrito sob um objetivo estendido. Os autores buscam minimizar o número de

realocações e o tempo de trabalho do guindaste. Para isso, as distâncias entre as colunas são consideradas. Na primeira fase, uma solução inicial factível é gerada heurísticamente através de uma regra. A segunda fase reduz o número de realocações utilizando um modelo de programação binária inteira. A terceira fase reduz o tempo de trabalho do guindaste utilizando um modelo de programação inteira mista. Os resultados dos testes numéricos mostraram que a heurística foi capaz de resolver instâncias com mais de 700 contêineres em um pouco mais de dois minutos.

(TANG *et al.*, 2012) lidam com o problema de alocação de placas, que é modelado como um problema de alocação de espaço de armazenamento de contêineres. Seguindo a estrutura do modelo de (WAN *et al.*, 2009), o objetivo é minimizar o custo das operações de movimentação (remanejamento e retirada), considerando a distância total percorrida pelo guindaste. O modelo é programado em Visual C++ 6.0 e a solução para instâncias de pequeno porte é obtida pelo CPLEX 11.0. Para resolver instâncias maiores é fornecido um algoritmo heurístico de horizonte rolante. A ideia de um horizonte rolante é dividir um problema de longo prazo em muitos subproblemas de curto prazo que são resolvidos sequencialmente, implementando soluções parciais em cada etapa. Para melhorar ainda mais as soluções heurísticas, desenvolve-se um método de pesquisa tabu com uma lista tabu de comprimento variável. Esta heurística é proposta para o caso em que a distância percorrida pelo guindaste pode ser ignorada. Experimentos numéricos mostram que os algoritmos heurísticos propostos e o método de pesquisa tabu são eficazes.

Duas formulações binário-inteiras diferentes para o problema são apresentadas em (CASERTA *et al.*, 2012). A principal diferença entre as duas formulações está na representação do horizonte de tempo. Na primeira formulação o horizonte de tempo é discreto e cada período de tempo é definido por um único movimento (de retirada ou remanejamento) e o problema é irrestrito. Enquanto isso, na segunda formulação, chamada de BRP-II, o problema é restrito e o horizonte de tempo é fixo, e definido como um conjunto de movimentos necessários para que um contêiner seja retirado do pátio. Em ambas as formulações, variáveis binárias indicam a posição do contêiner na área de armazenamento, se ele foi retirado ou realocado. O objetivo é minimizar o número total de realocações. É proposta uma regra heurística que ajuda a determinar a posição onde realocar um contêiner. A regra busca posições onde não haja um contêiner que vai sair antes do contêiner que está sendo remanejado, evitando assim mais remanejamentos no futuro. Caso não exista uma posição com essa característica disponível, a posição escolhida é a que leva ao remanejamento mais tardio. O algoritmo foi codificado em C++ e resolvido utilizando ILOG OPL 4.2. Em relação às formulações exatas, o *solver* não foi capaz de resolver instâncias maiores do que 4x7 dentro tempo de 86.400 segundos (um dia). Já para a heurística foram testadas instâncias de tamanho 10x10 e 100x100, cada

uma com duas configurações diferentes: com duas linhas da área de armazenamento vazia, ou com um terço de toda a área de armazenamento vazia.

(PETERING; HUSSEIN, 2013) introduzem o BRP-III, uma nova formulação matemática do BRP, onde variáveis binárias indicam a coluna na qual um bloco está e variáveis reais indicam sua posição relativa ao topo dessa pilha. Além disso, eles adicionam restrições de segurança, ao restringir que duas pilhas adjacentes nunca diferem em altura por mais de dois blocos. Comparando com o modelo BRP-I introduzido por (CASERTA *et al.*, 2012), o BRP-III tem consideravelmente menos variáveis inteiras. O modelo foi codificado em C++ usando o ILOG Concert Technology e testado em pequenas instâncias, de até 9 contêineres, usando o CPLEX 12.4. Para poder lidar com instâncias maiores, os autores introduzem um algoritmo chamado look-ahead (LA-N) que se baseia em duas ideias: (1) o algoritmo LA-N que foi desenvolvido independentemente por (CASERTA *et al.*, 2012) e (PETERING *et al.*, 2005) e (2) o conceito de movimento de limpeza introduzido por (AYDIN, 2006) e (ÜNLÜYURT; AYDIN, 2012). A heurística de LA é uma regra simples para limpar a área de armazenamento, alocando os blocos de forma que as realocações seguintes serão mínimas. A heurística LA-N permite movimentos de limpeza, que são relocações de um bloco que não está na pilha do bloco de destino. Após comparar seu método com cinco algoritmos diferentes para o BRP de (KIM; HONG, 2006), (AYDIN, 2006), (ÜNLÜYURT; AYDIN, 2012), (CASERTA *et al.*, 2012) e (LEE; LEE, 2010), os autores mostraram que o algoritmo LA-N tem o menor número de movimentos em todas as instâncias. A maior instância testada foi de tamanho 100x100. Além disso, concluiu-se que os movimentos de limpeza nem sempre são úteis quando o tamanho das instâncias aumenta porque a heurística executa muitos movimentos de limpeza, aumentando o número total de movimentos.

Também levando em consideração as propriedades do contêiner que será movido em seguida, (JOVANOVIĆ; VOß, 2014) propõe uma heurística chamada *chain heuristic*. Esta heurística é testada em instâncias de tamanho até 3x12, e mostrou resultados melhores do que a heurística proposta por (CASERTA *et al.*, 2012).

Em (ZEHENDNER *et al.*, 2015), o modelo BRP-II apresentado em (CASERTA *et al.*, 2012) é corrigido e são propostas algumas melhorias ao remover variáveis e parâmetros supérfluos, comprimindo algumas restrições e introduzindo novos limites superiores e inferiores. Além disso, é acrescentado um passo de pré-processamento para fixar várias variáveis. Esta nova formulação é chamada de BRP-II-A e é resolvida utilizando Cplex 12.5. Todas as instâncias de tamanho 5x4 foram resolvidas em tempo computacional razoável. Algumas instâncias com até 28 contêineres e dimensão de 6x6 e 5x7 puderam ser resolvidas. Já as instâncias de tamanho maior não tiveram solução encontrada, uma vez que o modelo ainda gera um número grande de variáveis e o processo de solução é

abortado devido à limitação de memória.

(EXPÓSITO-IZQUIERDO *et al.*, 2015) também propõem correções no modelo BRP-II apresentado em (CASERTA *et al.*, 2012) e desenvolvem um algoritmo baseado em *branch and bound*. Para reduzir o espaço de busca, apenas os nós mais promissores na árvore subjacente são explorados. Estes nós são escolhidos através do valor obtido em uma função que estima o custo de continuar explorando cada nó. Os resultados computacionais sugerem que o algoritmo porposto pode ser aplicado para resolver instâncias mais realistas.

(TRICOIRE *et al.*, 2018) abordam o CRP irrestrito. Os autores apresentam uma meta-heurística chamada *rake search*, inspirada em *beam search* e busca em árvore. Esta meta-heurística é uma busca em árvore incompleta: apenas certos sucessores são gerados em um determinado nó, até certo ponto. Depois disso, consiste em aplicar repetidamente diferentes heurísticas a diferentes soluções. Também são investigados vários fatores que influenciam algoritmos de *branch and bound* neste problema. Para os testes computacionais são utilizadas as instâncias de (CASERTA *et al.*, 2011).

(ZHU *et al.*, 2012) aplicam um algoritmo iterativo chamado *Iterative Deepening A** para o CRP restrito e irrestrito. Os efeitos de algumas heurísticas gulosas também são investigados. Também abordando o CRP irrestrito, mas com prioridades distintas e duplicadas, (TANAKA; TAKII, 2016) propõem um método exato baseado em *branch and bound* e utiliza novos *lower bounds*. Com isto, os autores encontraram resultados 1.848% melhores do que (ZHU *et al.*, 2012).

(GALLE *et al.*, 2018) propuseram um novo modelo de programação inteiro binário para o CRP restrito. O problema é formulado usando uma versão melhorada da codificação binária introduzida por (CASERTA *et al.*, 2009), na qual, ao invés de utilizar variáveis binárias que representam a configuração dos contêineres em um pátio, os autores utilizam uma única matriz binária. Com esta matriz é possível extrair todas as informações relacionadas à posição dos contêineres. O modelo é resolvido pelo Gurobi 7.0.1. Apesar de a formulação proposta reduzir o tamanho do problema, o modelo foi capaz de resolver todas as instâncias de tamanho 5x7, mas apenas algumas das instâncias de tamanho 6x6. Comparando com os resultados da literatura, esta formulação obteve melhores resultados do que diversas outras formulações e métodos exatos, exceto o *branch and bound* proposto por (TANAKA; TAKII, 2016).

Mais recentemente, (QUISPE *et al.*, 2018) implementaram o algoritmo *Iterative Deepening A** com novos *lower bounds*, um puramente combinatório e outro derivado da criação de bases de dados de padrões. Com isso, 4.1% mais instâncias foram resolvidas na otimalidade, quando comparado com os resultados obtidos por (ZHU *et al.*, 2012) e (TANAKA; TAKII, 2016).

(SILVA *et al.*, 2018a) lidam com um caso especial do CRP, onde existem apenas dois grupos de contêineres: o grupo-alvo a ser retirado primeiro e o restante dos contêineres seguintes. Além disso, um objetivo secundário é incluído, a minimização do número esperado de realocações para retirar os contêineres do próximo cliente. Os autores provam que tanto o problema com um único objetivo (reduzir o número de remanejamentos) quanto o problema biobjetivo são NP-Hard. Para resolver o problema biobjetivo é utilizado um algoritmo de *branch and bound* e um algoritmo *beam search*. Para o problema com um único objetivo é utilizado um algoritmo de *branch and bound* e uma heurística de tempo linear. Os autores comparam seus resultados computacionais com os resultados de (TANAKA; TAKII, 2016) e mostram que sua abordagem é mais rápida para resolver instâncias agrupadas.

Outro problema associado ao CRP é o problema do *Pre-marshalling*. Este problema ocorre quando os contêineres dentro de uma área de armazenamento devem ser rearranjados de forma que possam ser retirados posteriormente sem qualquer remanejamento adicional. A diferença entre este problema e o CRP é que no *Pre-marshalling* nenhum contêiner é retirado do pátio. A vantagem do *Pre-marshalling* é que os contêineres podem ser retirados rapidamente quando preciso, especialmente em situações na quais é preciso retirar muitos contêineres em um curto período de tempo. Entretanto, segundo (MONACO *et al.*, 2014b), em alguns casos pode ser que o *Pre-marshalling* não seja factível nem rentável. Isto depende do tempo e dos recursos disponíveis para realocar contêineres antes do início do carregamento do navio.

Sobre o problema do *Pre-Marshalling*, (LEE; HSU, 2007) apresenta um modelo de programação inteiro que tem um problema de fluxo de rede multi-commodity embutido. Os autores discutem diversas variações possíveis do modelo que podem ser obtidas modificando algumas das restrições, e propõem uma heurística para resolvê-lo de maneira mais eficiente. O modelo proposto e algumas variações são testadas em um exemplo com 14 contêineres em um pátio com 6 colunas e altura máxima de 4 contêineres. Para demonstrar a heurística proposta, um exemplo com 45 contêineres é testado. Bons resultados são obtidos, mas ainda a natureza complexa do problema limita as aplicações a problemas de pequena escala. (SILVA *et al.*, 2018b) apresentam um modelo unificado de programação inteira para resolver o *Pre-Marshalling* e o CRP nas variantes restrita e irrestrita. Uma heurística gulosa simples foi implementada para o *Pre-Marshalling* e para o CRP restrito foi implementado um algoritmo de *branch and bound*, que forneceu limites superiores para o problema irrestrito.

O problema do *Pre-marshalling* não é considerado neste trabalho.

Como um resumo dos trabalhos citados, pode-se dizer que a maioria adota como função objetivo a minimização do número total de relocações. Alguns outros traba-

lhos buscam minimizar o tempo de trabalho dos guindastes, como em (LEE; LEE, 2010; TANG *et al.*, 2012; ÜNLÜYURT; AYDIN, 2012; GALLE *et al.*, 2018; FIRMINO *et al.*, 2019). Os tipos de formulação utilizada pelos trabalhos que apresetaram modelos exatos são, em geral, programação linear inteira, programação inteira mista ou programação binária linear.

Como métodos de solução, além dos artigos citados anteriormente que utilizam heurísticas e meta-heurísticas, métodos exatos também tem sido amplamente utilizados, como os algoritmos baseados em A^* e *branch and bound*. O *solver* mais utilizado é o Cplex. Apesar disso, tem-se (WAN *et al.*, 2009) que utilizou o Xpress, e (GALLE *et al.*, 2018) que utilizou o Gurobi.

Uma visão geral da literatura sobre problemas de carregamento, descarregamento e *Pre-marshalling* é fornecida por (LEHNFELD; KNUST, 2014). Complementarmente, (STEENKEN *et al.*, 2004) descrevem e classificam os principais processos logísticos e operacionais em terminais de contêineres e apresentam um levantamento dos métodos utilizados para a otimização de tais problemas. Este levantamento é estendido e atualizado por (STAHLBOCK; VOSS, 2008). Uma revisão acerca dos principais problemas de tomada de decisão para a operação de terminais de contêineres é fornecida por (KIM; LEE, 2015). Em (CARLO *et al.*, 2014) uma revisão da literatura sobre operações de armazenamento em pátios de terminais de contêineres é feita, e um esquema formal de classificação destas operações é proposto. (LEE; SONG, 2017) fornecem uma visão ampla de vários problemas de transporte marítimo de contêineres, explicando as principais questões de planejamento estratégico, planejamento tático e gerenciamento de operações, que são categorizados em seis áreas de pesquisa.

3.3 O Problema do Plano de Estiva para Múltiplos Portos

O problema do plano de estiva para múltiplos portos (MPSP - *Multi-Port Stowage Planning*) consiste em determinar como organizar um determinado conjunto de contêineres de diferentes tipos em um conjunto de locais disponíveis em um navio-contêiner a fim de minimizar o número de movimentos desnecessários ao longo de uma rota de P portos.

A elaboração do plano de estiva está relacionada com a estrutura celular que o navio-contêiner possui, tal como ilustrado na Figura 3.7.

Observe na Figura 3.7 que o espaço da carga em um navio-contêiner é composto de células de 20 pés de comprimento. Cada coluna é uma composição de duas pilhas de unidades equivalentes de vinte pés (TEU - *Twenty foot Equivalent Unit*), e por isso,

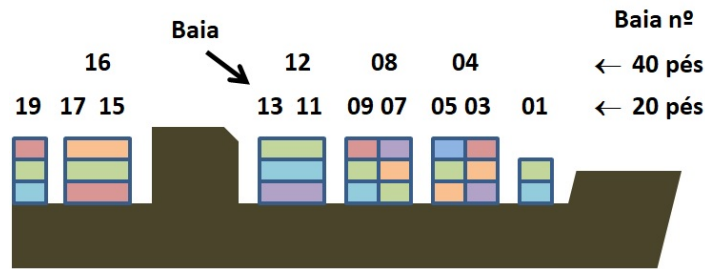


Figura 3.7 – Estrutura do local de armazenamento de um navio-contêiner. Adaptado de (BIERWIRTH; MEISEL, 2010).

contêineres de 40 TEU ocupam duas colunas de 20 pés. A identificação das colunas de 20 pés é feita por números ímpares, e a identificação das colunas de 40 pés é a média da numeração das colunas de 20 pés correspondente.

Uma vista transversal da baia 12 com suas pilhas verticais é dada na Figura 3.8. Cada número em uma célula indica o porto de destino do contêiner na posição correspondente.

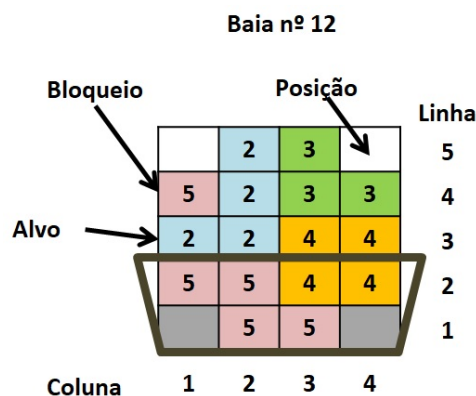


Figura 3.8 – Vista em corte transversal de uma baia. Adaptado de (BIERWIRTH; MEISEL, 2010).

Na Figura 3.8 é possível observar uma célula com um contêiner cujo destino é o porto 5 na posição $(Linha, Coluna) = (4, 1)$. Este contêiner está bloqueando o contêiner alvo que deveria ser descarregado no porto 2 e que ocupa a posição $(Linha, Coluna) = (3, 1)$.

Essa estrutura implica que um contêiner só pode ser movido se não houver outros contêineres acima dele. Caso contrário, esses contêineres de bloqueio devem ser retirados do navio para permitir o acesso ao contêiner alvo. Esse movimento é conhecido como remanejamento e deve ser minimizado para melhorar a eficiência do porto.

Movimentos de remanejamento podem ocorrer com frequência e levar a um maior tempo de permanência do navio no berço. Para evitar esse inconveniente é necessário elaborar o plano de estiva de modo que a decisão em um porto não implique muitos

movimentos de remanejamento nos próximos portos a serem visitados.

Para (SCIOMACHEN; TANFANI, 2007), p. 1434: “Formalmente, o problema de planejamento do plano de estiva consiste na determinação de como alocar um conjunto C de n contêineres de diferentes tipos em um conjunto S de m locais disponíveis em um navio-contêiner, levando em consideração algumas restrições estruturais e operacionais, relacionadas tanto aos contêineres, quanto ao navio, que são dados pelo tempo total necessário de estiva, que é o tempo necessário para o carregamento de todos os contêineres a bordo do navio mais o custo dos remanejamentos.”

Em (AVRIEL *et al.*, 2000), o MPSP é provado pertencer à classe de problemas NP-Completo mostrando que o mesmo pode ser transformado no problema de coloração de grafos circulares. Da mesma forma que o BRP, isso significa que não são conhecidos algoritmos de tempo polinomial para resolver o MPSP.

Consequentemente, devido à sua intratabilidade computacional para grandes instâncias, métodos heurísticos e meta-heurísticos são predominantemente utilizados para obtenção de soluções em tempo polinomial, como pode ser visto nestas abordagens na literatura: (BOTTER; BRINATI, 1991; AVRIEL *et al.*, 1998; WILSON; ROACH, 1999; WILSON; ROACH, 2000; DUBROVSKY *et al.*, 2002; AMBROSINO *et al.*, 2006; IMAI *et al.*, 2006; SCIOMACHEN; TANFANI, 2007; AMBROSINO *et al.*, 2010; FAN *et al.*, 2010).

Para (LEHNFELD; KNUST, 2014), os principais métodos desenvolvidos para resolver o problema do plano de estiva são programação dinâmica, abordagens heurísticas, programação inteira, algoritmo genético, *branch and bound* e *tabu-search*. (WILSON *et al.*, 2001) também adiciona à lista simulação baseada em métodos probabilísticos, sistemas especialistas baseados em regras e sistemas de suporte à decisão. Um levantamento do estado da arte dos métodos utilizados no problema de carregamento de navios — que é parte do problema do plano de estiva, é dado por (IRIS; PACINO, 2015).

Não obstante, alguns trabalhos apresentaram modelos exatos para o plano de estiva, entre eles temos: (CHEN *et al.*, 1995; AVRIEL *et al.*, 1998; KANG; KIM, 2002; AMBROSINO *et al.*, 2006; IMAI *et al.*, 2006; PACINO *et al.*, 2011; MONACO *et al.*, 2014a).

Em (AVRIEL *et al.*, 1998), o MPSP é formulado como um problema de programação linear binária que traz a solução para uma única baía de um navio que percorre um dado número de portos, assumindo que o número de contêineres que serão embarcados, junto com a sua origem e portos de destino, são conhecidos antecipadamente. o objetivo é minimizar o número de remanejamentos, e restrições de estabilidade do navio não são consideradas. Apesar de a formulação ser bastante concisa, a resolução de instân-

cias de tamanho real ainda é inviável devido ao alto número de variáveis e restrições que seriam geradas. Por isso, é desenvolvida uma heurística chamada de "*Suspensory Heuristic*", que atribui de forma dinâmica, espaços para alocar os contêineres. É utilizada pela heurística a regra de decisão "Remanejamento Míope" (*Myopic Shifting Rule*), que decide sobre remanejos acerca de uma única coluna. Para selecionar a coluna que deve sofrer um remanejamento, são utilizados dois métodos, o primeiro é uma função que seleciona preferencialmente colunas relativamente vazias, e o segundo, o "Remanejamento Necessário" (*Necessary Shift Rule*), é utilizado caso o movimento seja inevitável em mais de uma coluna e a coluna com menor índice esperado de remanejamentos em portos futuros deva ser selecionada. Finalmente, a regra do "Remanejamento Míope" é utilizada para decidir se os remanejamentos devem ser feitos no porto atual ou em portos futuros. A "*Suspensory Heuristic*" foi capaz de resolver um problema com 170 colunas, 10 linhas e 30 portos em 30 segundos. Os autores não apresentam resultados para o modelo exato.

(WILSON *et al.*, 2001), esboçam um sistema para gerar soluções para o problema de planejamento do plano de estiva. O método utilizado neste sistema define progressivamente o arranjo de contêineres dentro do espaço do navio até que cada contêiner seja individualmente alocado em uma posição, ou seja, o problema é dividido em dois subproblemas: um problema de colocação generalizada e um problema de colocação especializada. No processo de planejamento estratégico os contêineres generalizados são designados para um bloco de espaço e agrupados juntos. Para a resolução da primeira fase é utilizado o método *branch and bound*, e para a segunda fase é utilizada a busca tabu. Nos testes computacionais foram obtidos resultados generalizados em 90 minutos e soluções específicas em menos de uma hora.

(AMBROSINO *et al.*, 2004) apresentam um modelo de programação linear binária para o problema do plano de estiva. As variáveis indicam onde um contêiner é armazenado no navio. Nas restrições, o peso, tamanho e destino dos contêineres são considerados. Para carregar o navio, as autoras assumem que o navio começa sua rota no porto onde está sendo efetuado o carregamento e nos portos seguintes apenas as operações de descarregamento são permitidas, fazendo com que a estes portos não afetem o que acontece no navio, o que não é muito realista. Além disso, não é considerado explicitamente a possibilidade de remanejamentos associado ao carregamento. Para resolver um caso real de um porto em Gênova é proposta uma heurística que as permite relaxar algumas restrições do modelo e algumas regras de carregamento do navio são propostas. Esta heurística leva cerca de 20 minutos para elaborar um plano de estiva para navios com capacidade variando entre 138 TEU e 188 TEU.

(IMAI *et al.*, 2006) formulam um problema de programação inteira multi-objetivo. Eles estão preocupados em obter um plano de estiva que satisfaça restrições de

estabilidade do navio enquanto uma estimativa do número de remanejamentos de contêineres seja minimizada. Para resolvê-lo, uma heurística baseada em algoritmo genético é utilizada.

Uma variante do MPSP é o problema do plano de estiva para um único porto (SPSP - *single-port stowage planning problem*), que, de acordo com (PARREÑO-TORRES *et al.*, 2019), consiste em determinar o arranjo de contêineres no navio em um determinado porto sem considerar o carregamento de contêineres nos portos subsequentes.

Alguns desses trabalhos tem levado em consideração os atributos dos contêineres na elaboração do plano de estiva, como pode ser visto em (MONACO *et al.*, 2014b) e (IRIS *et al.*, 2018). Estes atributos, chamados de classes, são os seguintes: a dimensão (padrão, 45 pés, high-cube, superdimensionada), o peso (leve, médio, pesado), o tipo (refrigerado, topo aberto), a carga (perigosa, perecível) e o porto de destino.

(MONACO *et al.*, 2014b) abordam o problema do plano de estiva orientado ao terminal, ou seja, minimizando os custos relacionados ao pátio e às operações de transporte. Duas fases de planejamento do plano de estiva são detalhadas: Na primeira fase os contêineres são classificados com base em seus atributos, ou seja, os contêineres da mesma classe são armazenados em posições adjacentes, respeitando restrições para garantir o equilíbrio estático e dinâmico do navio. Nesta etapa, é considerado que a companhia marítima visa maximizar a utilização do navio e minimizar o número de remanejamentos a bordo. Um pré-plano de estiva é o *output* da primeira fase, que indica o contêiner exato a ser armazenado em cada posição do navio. Na segunda fase, o pré-plano de estiva atua como um dado de entrada e é transformado em um plano operacional, definindo uma lista de trabalho detalhada para cada guindaste do cais operar no navio. As restrições básicas desta etapa estão relacionadas à estabilidade do navio. No caso de um pré-plano de estiva detalhado, o operador do terminal portuário não tem como reduzir os custos operacionais, cabe a ele apenas aceitá-lo e eventualmente recorrer à política de *Pre-marshalling*. Um Modelo Inteiro Binário é proposto para a minimização dos tempos de transporte dos contêineres e dos remanejamentos no pátio. Entretanto, os autores ressaltam que os remanejamentos no pátio não podem ser calculados com exatidão. Restrições para assegurar que o peso dos contêineres em uma mesma coluna não aumente de baixo para cima e que regulam o peso máximo permitido em cada coluna do navio são impostas. Um algoritmo de *tabu search* de duas etapas é desenvolvido. A primeira etapa constrói uma solução inicial factível, e a segunda etapa procura por soluções factíveis melhores. Para resolver o problema com instâncias reais, o CPLEX é utilizado. O *tabu search* é capaz de encontrar soluções ótimas ou quase ótimas em baixo tempo computacional em todas as instâncias de teste, mesmo quando o CPLEX não consegue encontrar uma solução factível dentro do tempo limite de uma hora.

(IRIS *et al.*, 2018) também lidam com o problema de carregamento de contêineres em um único porto, assumindo como entrada um pré plano de estiva baseado em classes. O problema é integrado ao problema de *scheduling* de veículos de transferência, e o objetivo é reduzir os tempos de serviço dos equipamentos de manuseio e cumprir os prazos no tempo de carregamento do navio. Assume-se que um problema de *Pre-marshalling* foi resolvido, logo cada contêiner no pátio está pronto para ser retirado. Além disso, presume-se que as questões de estabilidade do navio tenham sido asseguradas pelo pré plano de estiva. As posições dos contêineres no pátio não são detalhadas, ao invés disso, uma variação no tempo de viagem dos veículos é adotada para estimar/simular os contêineres sendo armazenados em diferentes posições. Um modelo para obter limites inferiores para o problema é proposto, mas é intratável para instâncias de grande escala. Por causa disso, um GRASP é proposto. As instâncias testadas vão de 60 a 1000 contêineres para serem carregados no navio, e os resultados mostram que a heurística GRASP encontra soluções viáveis para todas as instâncias, com um gap médio de 10,9%.

(MEISEL; WICHMANN, 2010) apresentam uma metodologia para o planejamento da operação dos guindastes do cais para uma baía de um navio, considerando que as operações de carregamento e descarregamento são alternadas. Também são permitidos remanejamentos internos no navio. Os contêineres de remanejamento geralmente são descarregados do navio, armazenados em uma área intermediária no pátio do terminal portuário e, posteriormente, recarregados, de modo que estejam novamente a bordo no momento da partida. Os autores consideram que todos os contêineres pertencem à mesma classe, e por isso podem ser alocados em qualquer posição disponível para remanejamentos dentro do navio. A ideia defendida é de que o serviço pode ser acelerado com a permissão de remanejamentos internos. O objetivo é encontrar uma sequência de movimentação viável que converta uma determinada configuração de chegada de um navio em uma determinada configuração de partida dentro do tempo mínimo de serviço de um guindaste. Restrições de estabilidade no navio e uma tampa de escotilha não são consideradas. Além disso, os contêineres são diferenciados apenas pelo seu destino, que pode ser importação, exportação, remanejamento e contêineres fixos. A movimentação é restrita. Um modelo matemático de otimização foi formulado; portanto, instâncias de tamanho prático não podem ser resolvidas de forma ideal por solucionadores padrão dentro de um tempo razoável. Por causa disso, um GRASP foi projetado. A ideia do GRASP é construir uma solução viável em uma fase de construção da solução inicial e melhorar essa solução em uma fase de busca local, onde a randomização é aplicada dentro da fase de construção para gerar não apenas uma única solução, mas várias soluções diferentes. Testes computacionais foram conduzidos em instâncias de até 10, 15 e 20 pilhas e níveis. Os resultados mostram que, em média, cerca de 77% das operações de carga e descarga

podem ser realizadas em ciclos duplos e que 64% dos contêineres em uma baía podem ser remanejados internamente.

A próxima seção detalha e justifica a integração dos problemas do MPSP e CRP.

3.4 Integração

A ideia de integrar diferentes problemas de decisão é um tema de interesse na literatura porque leva em consideração o impacto das decisões para: Primeiro, toda a cadeia de operações ao longo do tempo e não apenas o impacto local para um porto, e; segundo, a cooperação entre os dois agentes mais representativos da logística portuária: a companhia marítima e o operador do terminal. Para (WILSON; ROACH, 2000), a eficiência do navio-contêiner é amplamente determinada pelo arranjo dos contêineres tanto dentro do terminal quanto no navio.

(MONACO *et al.*, 2014b) definem os diferentes decisores das companhias marítimas e os do terminal da seguinte forma: Os primeiros tem uma visão completa tanto dos contêineres a serem carregados e descarregados em cada porto durante a sua viagem, quanto a estrutura celular do navio. Já os segundos consideram apenas os contêineres a serem carregados naquele terminal em específico. Para os autores supracitados, os objetivos de cada planejador convergem. Para as companhias marítimas, os objetivos são maximizar a utilização do navio e minimizar o número de remanejamentos a bordo do navio, enquanto para o terminal os objetivos são minimizar o tempo de permanência que o navio fica atracado no berço, assim como os custos operacionais do terminal.

Neste sentido, existem estudos na literatura que destacam a importância de conciliar interesses conflitantes de diferentes *stakeholders* portuários, a fim de melhorar o desempenho geral de toda a cadeia, como (HA *et al.*, 2019b) e (BROOKS; SCHELLINCK, 2013).

No caso da integração do CRP com o MPSP, do ponto de vista da companhia marítima, definir o plano de estiva, conhecendo a localização dos contêineres nos pátios, significa garantir operações de carregamento e descarregamento mais rápidas e precisas. Enquanto isso, do ponto de vista dos operadores dos terminais de contêineres, a integração destes problemas de decisão faz sentido para planejar seus recursos para a transferência de carga tanto nas operações à beira-mar quanto nas terrestres.

A principal característica que permite que ambos os problemas sejam tratados de forma integrada é que tanto o pátio quanto o navio possuem uma estrutura celular e podem ser representados como matrizes. Esta estrutura sugere que existem características

semelhantes entre esses problemas, como por exemplo a restrição de que um contêiner somente é acessível se estiver no topo da coluna na qual se encontra. Por este motivo, é possível desenvolver uma abordagem para resolver conjuntamente o MPSP e o CRP.

O desafio deste problema integrado é escolher uma posição exata para cada contêiner, dentre todas as posições possíveis, de modo que o número total de realocações durante toda a rota de um navio seja minimizado.

Sobretudo, pode-se dizer que o número total de movimentos para remover todos os contêineres do pátio e embarcá-los em seu navio de destino é apenas uma estimativa do tempo necessário que o navio passará atracado no berço do porto. Conseqüentemente, a implicação prática de integrar os dois problemas de decisão é aumentar os lucros tanto do terminal quanto da companhia marítima, uma vez que o plano de estiva feito em conjunto com os portos pode levar a uma estimativa de tempo de permanência do navio no porto mais precisa, evitando assim custos extras por atrasos, abandono de carga ou aplicação de multas. Do lado do porto, menos movimentos feitos para carregar um navio significam economia de tempo e recursos utilizados, que pode ser contabilizado monetariamente.

Por estes motivos, não é coincidência de que a literatura mostre uma tendência de desenvolvimento em termos de integrar os principais problemas que aparecem em portos. Por exemplo: a alocação de berços integrada à alocação de guindastes (BIERWIRTH; MEISEL, 2010; CHANG *et al.*, 2010; IMAI *et al.*, 2008; MEISEL; BIERWIRTH, 2009; YANG *et al.*, 2012), alocação de berços e planejamento da operação do pátio, (HENDRIKS *et al.*, 2013), e alocação de contêineres vazios no pátio integrado ao roteamento de veículos (BRAEKERS *et al.*, 2013).

Esta tendência é justificada pelo fato óbvio de que a otimização de operações para apenas uma etapa não aumenta a eficiência geral do porto, uma vez que os estágios adicionais e não otimizados se comportam como bloqueio para o crescimento da eficiência portuária.

No entanto, embora a integração portuária seja um tópico de pesquisa há anos, a prática de fazer planejamento separado ou simplificado ainda tende a existir. Alguns exemplos são: focar no fluxo de exportação no problema de carregamento de um navio portuário e ignorar que o arranjo de contêineres é feito em pilhas no pátio ((MONACO *et al.*, 2014b; IRIS; PACINO, 2015; IRIS *et al.*, 2018)), considerar apenas o arranjo de carga no navio ((PARREÑO-TORRES *et al.*, 2019)), observar apenas o arranjo no pátio ((RAHMAN *et al.*, 2016)), ou ainda considerar o fluxo de transbordo, mas ignorando a estrutura de empilhamento para o navio e o pátio ((ZHEN *et al.*, 2016)).

Em relação a integração do CRP com o MPSP, nota-se que a literatura aborda apenas o carregamento do navio em um único porto, sem levar em consideração o impacto

da decisão nos portos seguintes, como pode ser visto em (JI *et al.*, 2015), (JOVANOVIĆ *et al.*, 2018) e (TANAKA; VOß, 2019). Este problema pode ser chamado de CRPSP - (*Container Relocation Problem with Stowage Planning*).

No CRPSP é assumido que o plano de estiva (posição exata de cada contêiner no navio) é dado como conhecido, mas a sequência de retirada deve ser decidida de forma que o número de realocações necessárias para mover todos os contêineres de uma baia do pátio para os locais designados em uma baia do navio seja minimizado.

(JI *et al.*, 2015) apresentam um modelo matemático para o CRPSP sob operações paralelas de múltiplos guindastes do cais. Em uma primeira etapa, um algoritmo genético, chamado de GA-ILSRS, decide a ordem de retirada dos contêineres do pátio, dado um plano de estiva conhecido. Na segunda etapa, dada a sequência de retirada obtida pelo GA-ILSRS, três estratégias heurísticas são apresentadas para escolher uma posição para um contêiner que precise ser remanejado, visando minimizar o número de remanejamentos; são elas: a estratégia da pilha mais próxima, a estratégia da pilha mais baixa e a estratégia de otimização. A última estratégia consiste em selecionar a pilha que não contém o próximo contêiner a ser retirado para realocar o contêiner que está sendo remanejado. O problema abordado é do tipo restrito. Diferentes classes de contêineres não são consideradas e é assumido que dentro de uma baia, todos os contêineres tem o mesmo tamanho. Instâncias de até 980 contêineres foram testadas. Quando comparado com o *branch and bound* proposto por (KIM; HONG, 2006), o número de remanejamentos do GA-ILSRS é superior. Entre as estratégias, os resultados dos experimentos mostram que a estratégia de otimização é a melhor.

(JOVANOVIĆ *et al.*, 2018) constroem um algoritmo guloso de dois passos e um GRASP para o CRPSP restrito. As mesmas instâncias de (JI *et al.*, 2015) são utilizadas, e os resultados mostraram que a abordagem proposta por (JOVANOVIĆ *et al.*, 2018) tem uma performance superior do que a proposta por (JI *et al.*, 2015).

(TANAKA; VOß, 2019) introduzem um algoritmo exato para o CRPSP nas versões restrita e irrestrita: um *branch and bound* com aprofundamento iterativo. A fim de restringir o espaço de busca e melhorar a eficiência, são propostas regras. Experimentos computacionais indicaram que o *branch and bound* com *lower bound* baseado em um relaxamento do problema original é capaz de resolver na otimalidade instâncias de pequeno e médio porte em um tempo de computação razoável, mas é ineficiente para instâncias de grande porte.

Como pôde ser percebido, não foram encontrados na literatura trabalhos que abordem a integração do CRP com o MPSP. Os maiores problemas da não consideração do CRP-MPSP são a falta de estudos sobre como essa sequência de retirada dos contêineres

do pátio afeta a organização dos contêineres no navio nos portos seguintes, gerando ou não futuros movimentos desnecessários; e a não consideração do espaço limitado do pátio durante a resolução do MPSP. Esta lacuna na literatura motivou o desenvolvimento deste trabalho.

Dado isto, para não deixar dúvidas sobre a importância da integração do MPSP e CRP para garantir uma operação eficiente entre o navio e todos os portos que este visita, um exemplo numérico é apresentado a seguir.

Para isso, primeiro é importante entender detalhadamente como as operações do navio e as áreas do pátio estão conectadas através do fluxo de contêineres, como ilustrado na Figura 3.9.

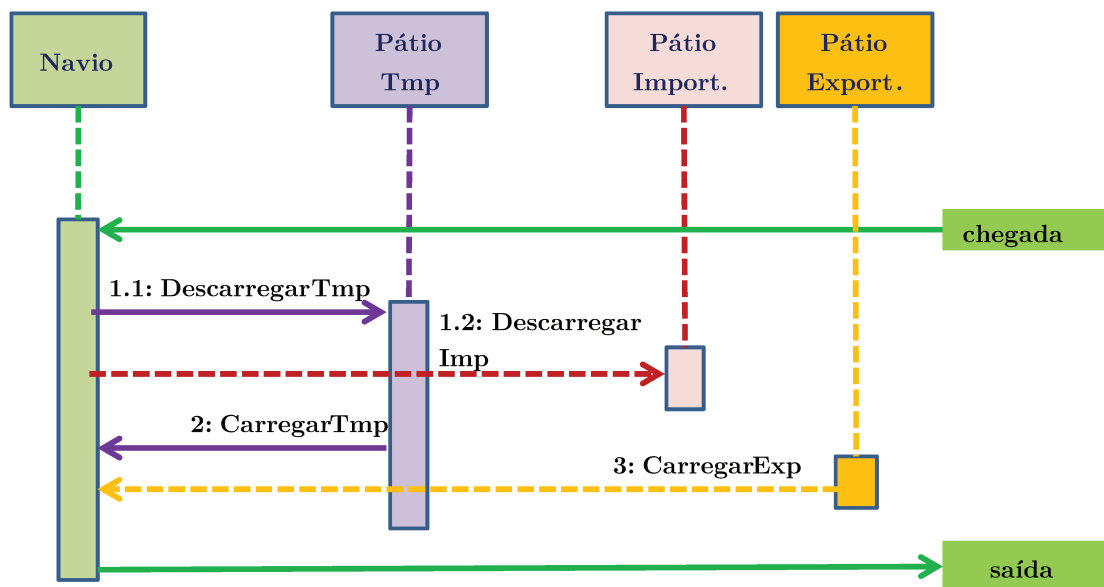


Figura 3.9 – Fluxo de contêineres entre as partes do navio e do pátio e suas operações correspondentes.

A Figura 3.9 emprega um diagrama de seqüência para mostrar como cada objeto colaborará entre si. A seqüência de operações começa com a chegada do navio ao porto. Então o navio começa a descarregar os contêineres para a área temporária (operação 1.1: Descarregar Tmp na Figura 3.9) ou para a área dos contêineres de importação (operação 1.2: Descarregar Navio na Figura 3.9) no pátio do porto. Essas duas operações são assíncronas, já que um contêiner pode ser descarregado alternadamente para área temporária ou de importação até que a operação de descarregamento dos contêineres que estão no navio e tem como destino este porto seja concluída.

Após as operações de descarregamento do navio, o segundo passo é recarregar o conjunto de contêineres de remanejamento que foram temporariamente armazenados na área temporária do pátio (operação 2: Carregar Temp na Figura 3.9).

O terceiro passo é escolher uma sequência adequada para carregar um a um os contêineres de exportação do pátio no navio (operação 3: Carregar Exp na Figura 3.9). Essa sequência deve procurar a minimização do número de movimentos desnecessários no pátio de exportação e dentro do navio. Após o término deste passo, o navio pode partir para o próximo porto.

As operações 2 e 3 podem ser efetuadas tanto de forma assíncrona quanto síncrona.

Entendido esse sequenciamento, observe na Figura 3.10 que um navio chega ao segundo porto de sua rota e descarrega os contêineres que tinham como destino final este porto (operação 1.2). Como não havia nenhum contêiner bloqueando o acesso aos contêineres que deveriam ser descarregados, nenhum contêiner foi descarregado para a área temporária do pátio (operação 1.1). Nas Figuras 3.10, 3.11, 3.12 e 3.13 todos os números dos contêineres representam o porto de destino de cada contêiner que ocupa aquele espaço do navio ou do pátio.

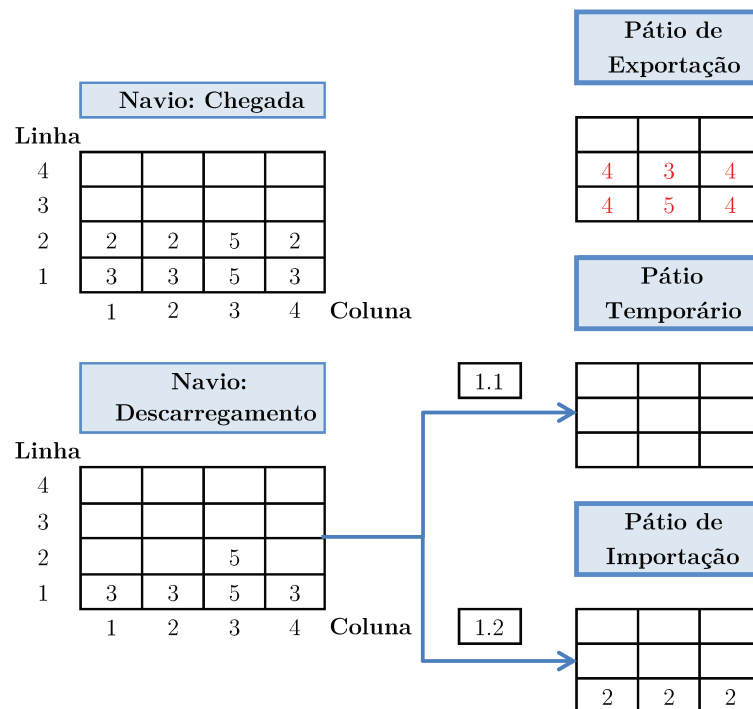


Figura 3.10 – Primeiras operações após a chegada do navio no porto 2: descarregamento.

A segunda operação a ser efetuada é o carregamento dos contêineres no pátio de exportação. Seguindo o exemplo da Figura 3.11, a sequência selecionada foi: 3, 5, 4, 4, 4, 4. Esta sequência de carregamento não causou nenhum remanejamento no pátio.

Embora o resultado final dessa escolha de movimentos não tenha causado nenhum remanejamento no porto 2, observe que o navio parte com quatro contêineres cujo destino é o porto 4 acima de dois contêineres cujo destino é o porto 3. Isso significa

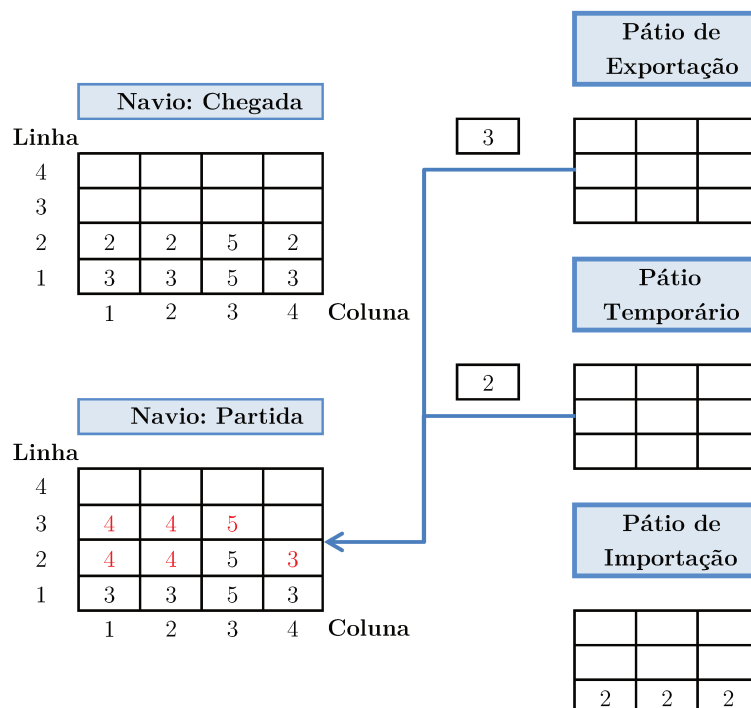


Figura 3.11 – Carregamento no navio dos contêineres do pátio de exportação do porto 2.

que, no porto 3, o navio terá que efetuar quatro movimentos de remanejamento para poder completar o descarregamento.

Uma alternativa à seleção de movimentos seleciona nas Figuras 3.10 e 3.11 é a política 2, que poderia evitar dois movimentos adicionais no porto 3, conforme descrito nas Figuras 3.12 e 3.13.

Observe na Figura 3.12 que o mesmo navio chega ao segundo porto de sua rota e descarrega os contêineres que tinham como destino final este porto (operação 1.2) e também descarrega para a área temporária do pátio dois contêineres cujo destino final é o porto 3 (operação 1.1).

Consequentemente, é possível observar na Figura 3.13 que quando o navio parte do segundo porto depois de carregar os contêineres no pátio temporário (operação 2) e no pátio de exportação (operação 3), nenhum remanejamento adicional será necessário para efetuar o descarregamento no porto seguinte.

Este exemplo numérico mostrou que nem sempre a melhor sequência de movimentação em um porto é a melhor para toda a rota de um navio. Outro ponto que também foi esclarecido foi o que a mesma sequência de retirada dos contêineres do pátio de exportação poderia fornecer um número diferente de movimentos devido ao plano de estiva adotado pelo navio.

Esse tipo de situação acontece porque a organização dos contêineres em pilhas

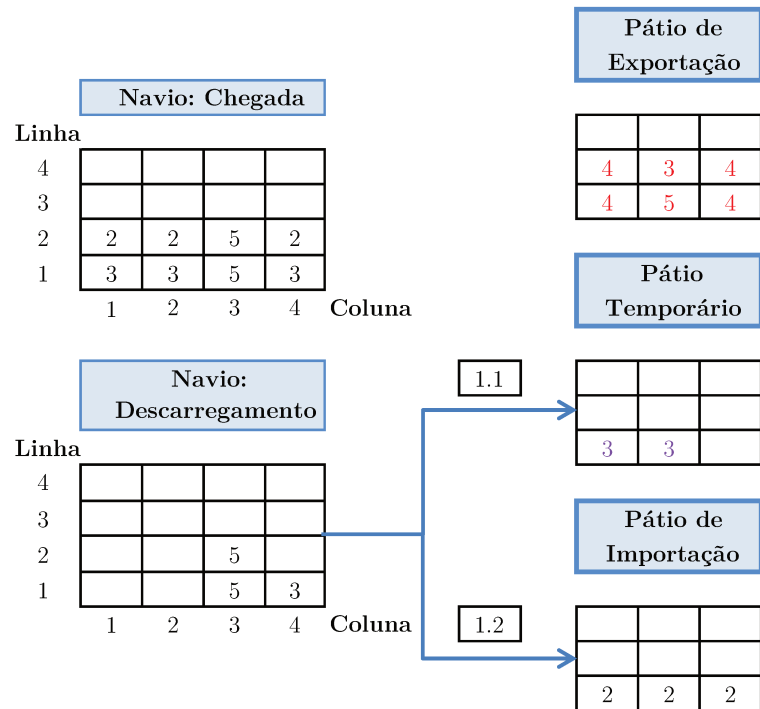


Figura 3.12 – Política alternativa à apresentada na Figura 3.10.

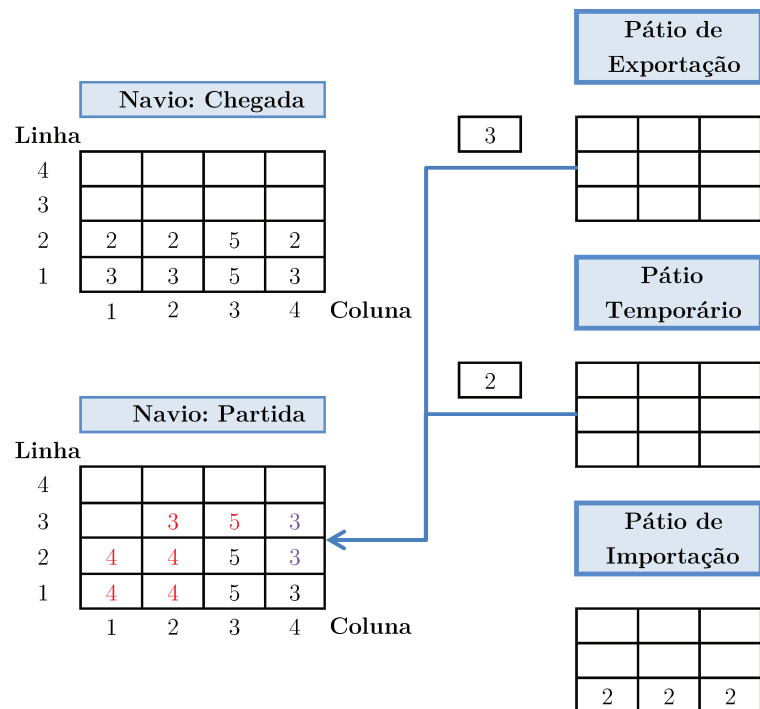


Figura 3.13 – Carregamento no navio dos contêineres do pátio temporário e de exportação do porto 2.

age como uma memória que conecta sucessivas decisões. Dessa forma, a decisão de planejamento de navios contêineres para cada porto depende parcialmente do que foi feito nos portos anteriores, uma vez que decisões anteriores não otimizadas de forma integrada

podem produzir contêineres de bloqueio.

Este exemplo numérico reforça a necessidade de empregar uma abordagem de planejamento integrado para obter soluções de melhor qualidade.

Assim sendo, o próximo capítulo detalha a modelagem do CRP-MPSP.

4 Modelagem do CRP-MPSP

O modelo do CRP-MPSP estende e integra os modelos apresentados em (ZEHENDER *et al.*, 2015) e (AVRIEL *et al.*, 1998). Estes dois modelos foram escolhidos dentro todos os outros apresentados no Capítulo 3 por serem modelos concisos e por permitirem a integração sem modificações complexas.

Como explicado anteriormente, temos que neste modelo é representada a viagem de um navio por uma rota de P portos. Em cada porto $p \in \{1, \dots, P-1\}$ em que este navio atracar existe um pátio com N_o contêineres de exportação que devem ser embarcados nele, sendo que $o \in \{1, \dots, P-1\}$ (veja a Figura 4.1). No primeiro porto $p = 1$ o navio chega vazio e recebe o carregamento dos contêineres deste porto que estavam no pátio. Nos portos $p \in \{2, \dots, P-1\}$, primeiro o navio descarrega os contêineres com destino ao porto p no qual se encontra, como ilustrado na Figura 4.2 e então recebe o carregamento de contêineres com destino aos portos seguintes (veja a Figura 4.3). Finalmente, no último porto P , apenas o descarregamento é efetuado.

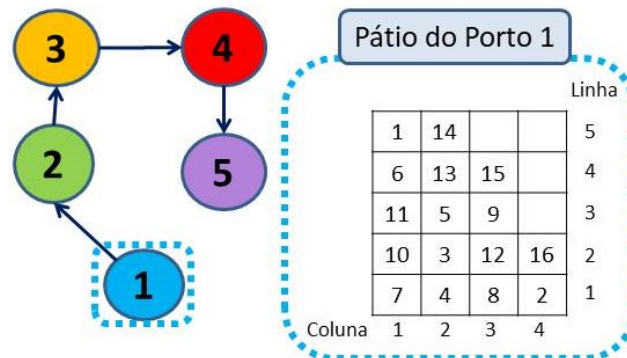


Figura 4.1 – Exemplo de uma rota de um navio por 5 portos.

Note que, para retirar todos os contêineres de cada pátio de cada porto p pertencente a rota do navio, um problema similar ao BRP, proposto por (CASERTA *et al.*, 2012), deve ser resolvido. A diferença é que no modelo integrado aqui proposto, a ordem de retirada dos contêineres não é imposta. A única informação exigida com antecedência pelo modelo é a localização de cada contêiner no pátio e seu destino. Desta forma, o modelo decide a sequência de remoção de cada contêiner em cada pátio portuário, observando toda a jornada do navio.

Note também que na Figura 4.1 os contêineres são representados por diferentes índices do que nas Figuras 4.2 e 4.3. Isso acontece porque, no pátio, os índices representam apenas uma identificação de cada contêiner, não a ordem de remoção. Enquanto isso, no navio, os índices representam o porto de destino de cada contêiner.

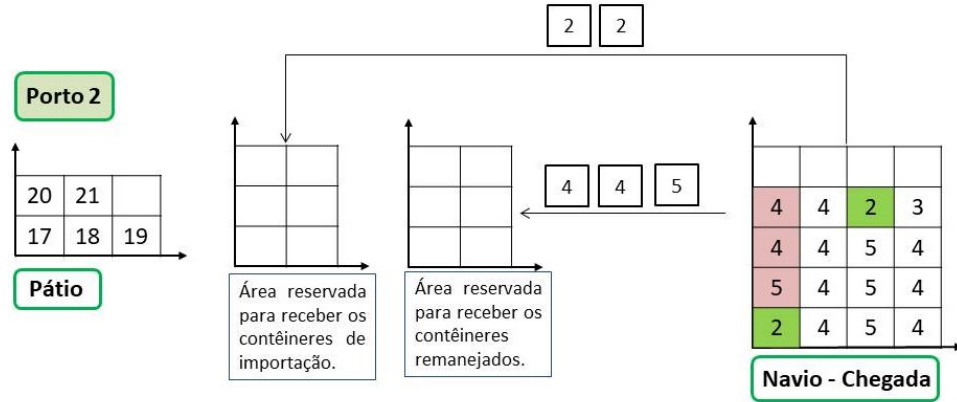


Figura 4.2 – Exemplo do descarregamento de um navio.

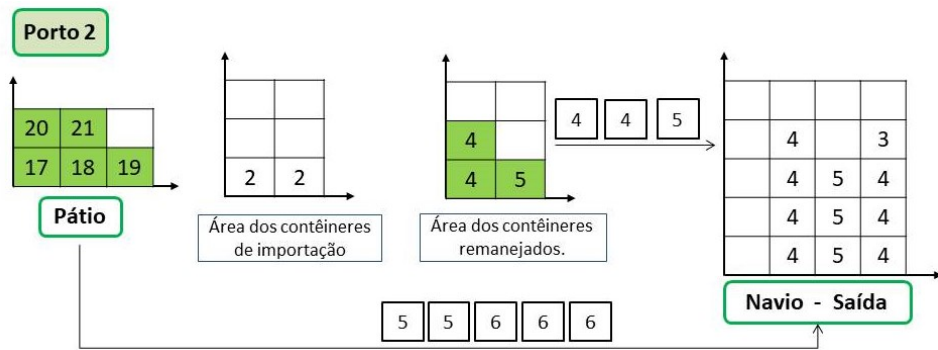


Figura 4.3 – Exemplo do carregamento de um navio.

Na Figura 4.2 podemos ver que o navio chega ao segundo porto de sua rota e tem que descarregar os dois contêineres destacados em verde, dos quais um deles não está no topo de sua coluna. Por esse motivo, primeiro os contêineres de bloqueio localizados acima do contêiner de destino serão descarregados e temporariamente alocados em uma área reservada do pátio. Em seguida, podemos ver na Figura 4.2 que após todos os contêineres destinados a este porto terem sido descarregados do navio, a operação de carregamento dos contêineres do segundo pátio portuário e dos contêineres remanejados é iniciada.

O modelo exato segue essa representação e é descrito detalhadamente na seção seguinte.

4.1 Modelo matemático do CRP-MPSP

As seguintes premissas foram feitas por questão de simplicidade, sem comprometer a aplicação geral:

- O navio tem um formato retangular e pode ser representado como uma matriz com linhas ($r \in \mathbf{R}$) e colunas ($c \in \mathbf{C}$), cuja capacidade máxima é de $R \times C$ contêineres.

Cada espaço para se armazenar um contêiner no navio é uma coordenada (r, c) , onde $r \in \{1, \dots, R\}$ e $c \in \{1, \dots, C\}$.

- O pátio também pode ser representado como uma matriz com linhas ($h \in \mathbf{H}_o$) e colunas ($w \in \mathbf{W}_o$). Cada espaço para se armazenar um contêiner no pátio é uma coordenada (i, j) , onde $i \in \{1, \dots, W\}$, $j \in \{1, \dots, H\}$ e $o \in \{1, \dots, P - 1\}$.
- Todos os contêineres tem o mesmo tamanho e peso e são diferenciados apenas pelo seu porto de destino.
- Todos os contêineres são acessíveis apenas pelo topo.
- O navio contêiner sempre pode carregar todos os contêineres disponíveis em cada porto e sua capacidade nunca será excedida.
- Em cada pátio o , para garantir a acessibilidade de todos os contêineres, deve haver pelo menos $H_o - 1$ espaços vazios, onde $W_o \times H_o \geq N_o + (H_o - 1)$ deve ser satisfeito.
- Cada período de tempo $t \in \mathbf{T}_o$, onde $o \in \mathbf{P}$, é definido como um conjunto completo de movimentos para se retirar um contêiner do pátio e carregá-lo no navio. Isto inclui movimentos de remanejamento, se necessário.
- Todos os contêineres dos pátios serão embarcados no navio.
- Os contêineres que são descarregados do navio são destinados para uma área específica do pátio e não são misturados com os contêineres que serão embarcados neste mesmo navio, como visto nas Figuras 4.2 e 4.3.

Vale a pena lembrar que um navio com formato irregular pode ser concebido simplesmente adicionando restrições que representam contêineres imaginários que ocupam os mesmos espaços durante toda a viagem (DING; CHOU, 2015). Apesar disso, o formato irregular não é considerado neste trabalho.

A seguir estão detalhados os parâmetros e conjuntos utilizados na modelagem.

Nomenclatura

Índices

$a \in \{o + 1, \dots, d\}$ Porto onde o contêiner n é remanejado.

$c \in \mathbf{C}$ Índice das colunas do navio.

$d \in \{o + 1, \dots, P\}$ Destino do contêiner n .

$i \in \mathbf{W}_o$ Índice das linhas dos pátios.

$j \in \mathbf{H}_o$ Índice das colunas dos pátios.

$k \in \mathbf{W}_o$ Índice das linhas dos pátios quando um remanejamento é efetuado.

$l \in \mathbf{H}_o$ Índice das colunas dos pátios quando um remanejamento é efetuado.

$n \in \mathbf{\Omega}_o$ Índice do contêiner.

$o \in \{1, \dots, P - 1\}$ Origem do contêiner n .

$p \in \mathbf{P}$ Índice do porto.

$r \in \mathbf{R}$ Índice das linhas do navio.

$t \in \mathbf{T}_o$ Índice dos períodos de tempo.

Parâmetros e Conjuntos

$\phi_{o,d}$ Contém o índice n de um contêiner dada a sua origem o e destino d .

θ_o Número total de contêineres no navio no momento em que ele sai do porto o .

$\mathbf{\Omega}_o$ Índices dos contêineres em cada pátio o .

\mathbf{C} Colunas do navio.

$\mathbf{F}_{o,d}$ Matriz de transporte, de dimensão $(P - 1) \times (P - 1)$, que contém o número de contêineres com origem o e destino d .

\mathbf{H}_o Linhas de cada pátio o .

\mathbf{M} Um número suficientemente grande.

\mathbf{N}_o Contêineres em cada pátio o .

R Linhas do navio.

T_o Períodos de tempo em cada pátio o .

W_o Colunas de cada pátio o .

4.1.1 Variáveis do Problema

As variáveis estão divididas em três grupos. O primeiro grupo é composto das variáveis exclusivas para o CRP, que são as variáveis de configuração e as variáveis de movimento. O segundo grupo é composto das variáveis de integração entre os dois problemas, e o terceiro grupo é composto das variáveis exclusivas para o plano de estiva.

- *variáveis de configuração*: Indicam onde os contêineres estão alocados no pátio em cada período de tempo t .

$$b_{ijnt} = \begin{cases} 1, & \text{se o contêiner } n \text{ está na posição } (i, j) \text{ no período de tempo } t, \\ 0, & \text{caso contrário;} \end{cases}$$

$$v_{nt} = \begin{cases} 1, & \text{se o contêiner } n \text{ foi retirado no período } t', \text{ onde } t' \in \{1, \dots, t-1\}, \\ 0, & \text{caso contrário;} \end{cases}$$

- *variáveis de movimento*: Variáveis que contam os movimentos no pátio, seja de retirada ou de remanejamento.

$$x_{ijklnt} = \begin{cases} 1, & \text{se o contêiner } n \text{ é realocado de } (i, j) \text{ para } (k, l) \text{ no período de tempo } t, \\ 0, & \text{caso contrário;} \end{cases}$$

$$y_{ijnt} = \begin{cases} 1, & \text{se o contêiner } n \text{ é retirado de } (i, j) \text{ no período de tempo } t, \\ 0, & \text{caso contrário;} \end{cases}$$

- *variável de carregamento do navio*: Integra a movimentação no pátio e no navio.

$$z_{ntrc} = \begin{cases} 1, & \text{se no período de tempo } t, \text{ o contêiner } n \text{ ocupa a posição } (r, c) \text{ no navio,} \\ 0, & \text{caso contrário;} \end{cases}$$

$$q_{odrc} = \begin{cases} 1, & \text{se um contêiner com destino final o porto } d \text{ é remanejado no porto } o \text{ e} \\ & \text{realocado na posição } (r, c) \text{ no navio,} \\ 0, & \text{caso contrário;} \end{cases}$$

- *variáveis do navio*: Variáveis que indicam a movimentação no navio.

$$w_{odarc} = \begin{cases} 1, & \text{se há um contêiner na posição } (r, c) \text{ que foi embarcado no porto } o, \\ & \text{com destino final no porto } d \text{ e remanejado porto } a, \\ 0, & \text{caso contrário;} \end{cases}$$

$$u_{orc} = \begin{cases} 1, & \text{se saindo do porto } o, \text{ a posição } (r, c) \text{ está ocupada por um contêiner,} \\ 0, & \text{caso contrário.} \end{cases}$$

Assim como em (AVRIEL *et al.*, 1998), os possíveis valores para o índice a são $o+1, o+2, \dots, d-1, d$. Se $a = d$, o contêiner é descarregado no seu destino final, enquanto que se $a < d$, o contêiner é remanejado, ou seja, é descarregado no porto o e recarregado novamente no mesmo porto. Ele será então denotado $w_{ada'r'c'}$, onde (r', c') pode ser uma posição diferente no navio.

4.1.2 Função Objetivo

O objetivo do modelo é minimizar o número total de remanejamentos, o que inclui o número total de remanejamento nos pátios (representado pelo somatório da variável x_{ijklnt} para todos os pátios $o = 1, \dots, P-1$), e o número total de remanejamentos no navio (representado pela somatório da variável w_{oadrc}).

$$\text{Minimizar } \sum_{o=1}^{P-1} \sum_{i=1}^{W_o} \sum_{j=1}^{H_o} \sum_{k=1}^{W_o} \sum_{l=1}^{H_o} \sum_{n \in \Omega_o} \sum_{t=1}^{T_o} x_{ijklnt} + \sum_{o=1}^{P-1} \sum_{d=o+1}^P \sum_{a=o+1}^{d-1} \sum_{r=1}^R \sum_{c=1}^C w_{oadrc} \quad (4.1)$$

4.1.3 Restrições

I: *Restrições dos pátios*

$$\sum_{i=1}^{W_o} \sum_{j=1}^{H_o} b_{ijnt} + v_{nt} = 1 \quad (4.2)$$

$$n \in \Omega_o; t \in T_o; o \in \{1, \dots, P-1\}$$

$$\sum_{n \in \Omega_o} b_{ijnt} \leq 1 \quad (4.3)$$

$$i \in W_o; j \in H_o; t \in T_o; o \in \{1, \dots, P-1\}$$

$$\sum_{n \in \Omega_o} b_{ijnt} \geq \sum_{n \in \Omega_o} b_{i(j+1)nt} \quad (4.4)$$

$$i \in W_o; t \in T_o; j \in \{1, \dots, H_o - 1\}; o \in \{1, \dots, P-1\}$$

$$b_{ijnt} = b_{ijn(t-1)} + \sum_{k=1}^{W_o} \sum_{l=1}^{H_o} x_{klijn(t-1)} - \sum_{k=1}^{W_o} \sum_{l=1}^{H_o} x_{ijkln(t-1)} - y_{ijn(t-1)} \quad (4.5)$$

$$n \in \Omega_o; i \in W_o; j \in H_o; t \in \{2, \dots, T_o\}; o \in \{1, \dots, P-1\}$$

$$v_{nt} = \sum_{i=1}^{W_o} \sum_{j=1}^{H_o} \sum_{t'=1}^{t-1} y_{ijnt'} \quad (4.6)$$

$$n \in \Omega_o; t \in \{2, \dots, T_o\}; o \in \{1, \dots, P-1\}$$

$$M \left(1 - \sum_{n \in \Omega_o} x_{ijklnt} \right) \geq \sum_{n \in \Omega_o} \sum_{j'=j+1}^{H_o} \sum_{l'=l+1}^{H_o} x_{ij'kl'nt} \quad (4.7)$$

$$i, k \in W_o; j, l \in H_o; t \in T_o; o \in \{1, \dots, P-1\}$$

$$M \left(1 - \sum_{j=1}^{H_o} b_{ijnt} \right) \geq \sum_{j=1}^{H_o} \sum_{k=1}^{W_o} \sum_{l=1}^{H_o} \sum_{n \in \Omega_o} \sum_{i' \neq i}^{W_o} x_{i'jklnt} \quad (4.8)$$

$$n \in \Omega_o; i \in W_o; t \in T_o; o \in \{1, \dots, P-1\}$$

$$x_{ijilnt} = 0 \quad (4.9)$$

$$n \in \Omega_o; i \in W_o; j, l \in H_o; t \in T_o; o \in \{1, \dots, P-1\}$$

$$\sum_{k=1}^{W_o} \sum_{l=1}^{H_o} \sum_{n \in \Omega_o} x_{i(j+1)klnt} - \sum_{n \in \Omega_o} b_{i(j+1)nt} + 1 \geq \sum_{k=1}^{W_o} \sum_{l=1}^{H_o} \sum_{n \in \Omega_o} x_{ijklnt} + \sum_{n \in \Omega_o} y_{ijn} \quad (4.10)$$

$$i \in W_o; j \in \{1, \dots, H_o - 1\}; t \in T_o; o \in \{1, \dots, P-1\}.$$

II: Restrições de integração

$$\sum_{n \in \Omega_o} v_{nt} = t \quad (4.11)$$

$$t \in \mathbf{T}_o; o \in \{1, \dots, P-1\}$$

$$v_{n(t+1)} \geq v_{nt} \quad (4.12)$$

$$n \in \Omega_o; t \in \{1, \dots, T_o - 1\}; o \in \{1, \dots, P-1\}$$

$$\sum_{r=1}^R \sum_{c=1}^C z_{ntrc} = v_{n(t+1)} \quad (4.13)$$

$$n \in \Omega_o; t \in \{1, \dots, T_o - 1\}; o \in \{1, \dots, P-1\}$$

$$\sum_{o'=1}^{o-1} \sum_{d=o+1}^P \sum_{a=o+1}^d w_{o'darc} + \sum_{d=o+1}^P q_{odrc} + \sum_{n \in \Omega_o} z_{ntrc} \leq 1 \quad (4.14)$$

$$r \in \mathbf{R}; c \in \mathbf{C}; t \in \mathbf{T}_o; o \in \{1, \dots, P-1\}$$

$$z_{n(t+1)rc} \geq z_{ntrc} \quad (4.15)$$

$$n \in \Omega_o; r \in \mathbf{R}; c \in \mathbf{C}; t \in \{1, \dots, T_o - 1\}; o \in \{1, \dots, P-1\}$$

$$\sum_{a=o+1}^d w_{odarc} = \sum_{n \in \phi_{o,d}} z_{nT_orc} + q_{odrc} \quad (4.16)$$

$$r \in \mathbf{R}; c \in \mathbf{C}; d \in \{o+1, \dots, P\}; o \in \{1, \dots, P-1\}$$

$$\sum_{r=1}^R \sum_{c=1}^C z_{nT_orc} = 1 \quad (4.17)$$

$$n \in \Omega_o; o \in \{1, \dots, P-1\}$$

$$\sum_{n \in \Omega_o} z_{ntrc} + \sum_{o'=1}^{o-1} \sum_{d=o+1}^P \sum_{a=o+1}^d w_{o'darc} + \sum_{d=o+1}^P q_{odrc} \geq \sum_{n \in \Omega_o} z_{nt(r+1)c} \quad (4.18)$$

$$t \in \mathbf{T}_o; r \in \{1, \dots, R-1\}; c \in \mathbf{C}; o \in \{1, \dots, P-1\}$$

$$\sum_{o'=1}^{o-1} \sum_{c=1}^C \sum_{r=1}^R w_{o'dorc} = \sum_{r=1}^R \sum_{c=1}^C q_{odrc} \quad (4.19)$$

$$d \in \{o+1, \dots, P\}; o \in \{1, \dots, P-1\}$$

III: Restrições do navio

$$\sum_{a=o+1}^d \sum_{r=1}^R \sum_{c=1}^C w_{odarc} - \sum_{m=1}^{o-1} \sum_{r=1}^R \sum_{c=1}^C w_{mdorc} = F_{od} \quad (4.20)$$

$$o \in \{1, \dots, P-1\}; d \in \{o+1, \dots, P\}$$

$$\sum_{m=1}^o \sum_{d=o+1}^P \sum_{a=o+1}^d w_{mdarc} = u_{orc} \quad (4.21)$$

$$o \in \{1, \dots, P-1\}; r \in \mathbf{R}; c \in \mathbf{C}$$

$$u_{orc} - u_{o(r+1)c} \geq 0 \quad (4.22)$$

$$o \in \{1, \dots, P-1\}; r \in \{1, \dots, R-1\}; c \in \mathbf{C}$$

$$\sum_{o=1}^{d-1} \sum_{e=d}^P w_{oedrc} + \sum_{o=1}^{d-1} \sum_{e=d+1}^P \sum_{a=d+1}^e w_{oea(r+1)c} \leq 1 \quad (4.23)$$

$$d \in \{2, \dots, P\}; r \in \{1, \dots, R-1\}; c \in \mathbf{C}$$

$$\sum_{c=1}^C \sum_{r=\lceil \theta_o / C \rceil + 1}^R u_{orc} = 0 \quad (4.24)$$

$$o \in \{1, \dots, P-1\}$$

As restrições (4.2) até (4.10) são exclusivas para o processo de retirada dos contêineres dos pátios nos portos onde o navio atracar. A restrição (4.2) garante que em cada período de tempo, cada contêiner deve estar dentro do pátio ou fora dele. A restrição (4.3) garante que em cada período de tempo, cada posição (i, j) deve estar ocupada por um único contêiner. A restrição (4.4) garante que não haja ‘buracos’ no pátio ao restringir que se há um contêiner posição $(i, j+1)$, então a posição (i, j) abaixo também deve estar ocupada. A restrição (4.5) é a restrição de equilíbrio de fluxo entre as variáveis de configuração e de movimento no pátio. Ele vincula o *layout* no período t com o *layout* no período $t-1$ através das retiradas e realocações executadas. A restrição (4.6) estabelece que todos os contêineres devem ser removidos do pátio. A restrição (4.7) garante a política LIFO. Essa restrição é necessária porque mais de um remanejamento pode ser executado por período de tempo. Ou seja, se no período t , o contêiner n_1 está abaixo do contêiner n_2 , e o contêiner n_1 é remanejado, então no período $t+1$ o contêiner n_1 não pode estar alocado em uma posição abaixo do contêiner n_2 . A restrição (4.8) garante que sejam remanejados apenas os contêineres que estão acima (na mesma coluna) do contêiner objetivo. A restrição (4.9) garante que nenhum contêiner pode ser remanejado para outra posição na mesma coluna na qual ele se encontra. A restrição (4.10) garante que um contêiner na posição (i, j) só pode ser movido depois que o contêiner na posição $(i, j+1)$ é movido. Se o contêiner na posição $(i, j+1)$ não é movido então temos que $b_{i(j+1)nt} = 1$ e $x_{i(j+1)klnt} = 0$, e o lado esquerdo da equação se torna 0. Consequentemente o lado direito da equação também deve ser 0. Dessa forma, nenhuma realocação ou remanejamento é permitido para o contêiner na posição (i, j) .

As restrições (4.11) até (4.19) são de integração entre o pátio e o navio. Estas restrições garantem que o navio seja preenchido de forma correta. A restrição (4.11) atua como um contador dos contêineres que saíram do pátio por período de tempo. Isto é, no período de tempo $t = 1$, um contêiner deve ter sido retirado do pátio, no período $t = 2$, dois contêineres devem ter sido retirados do pátio e assim por diante até que todos os N_o contêineres tenham sido retirados no período T_o . A restrição (4.12) garante que um contêiner seja retirado do pátio por período de tempo. Ou seja, quando um contêiner n é retirado do pátio, a variável v_{nt} se torna 1 e se mantém igual a 1 nos períodos de tempo seguintes. A restrição (4.13) garante que o contêiner n seja carregado no navio no período de tempo t . A restrição (4.14) assegura que uma posição (r, c) no navio só pode ser ocupada por um único contêiner, seja ele um contêiner que foi carregado no porto atual (porto o), em algum porto anterior (porto $o - 1$) ou um contêiner que já estava no navio e está sendo remanejado em o . A restrição (4.15) certifica que o contêiner n , depois de carregado, não mude de posição enquanto o navio estiver parado no mesmo porto. A restrição (4.16) garante que se há um contêiner na posição (r, c) do navio, ele deve ser um contêiner que acabou de ser embarcado, ou um contêiner de remanejamento. A restrição (4.17) assegura que todos os N_o contêineres do pátio do porto o já foram embarcados no navio. A restrição (4.18) garante que, durante o processo de carregamento do navio, nenhum contêiner seja alocado em uma posição flutuante ou que ocupe a posição de um contêiner que já estava no navio ou foi remanejado. A restrição (4.19) contabiliza o número total de contêineres que foram remanejados no porto o .

As restrições (4.20) até (4.24) são exclusivas para o navio. A restrição (4.20) garante a conservação de fluxo, e indica que o número total de contêineres no porto o deve ser igual ao número de contêineres que foram embarcados nos portos $p = \{1, 2, \dots, o\}$ menos os contêineres que foram desembarcados nos portos $p = \{1, 2, \dots, o\}$. A restrição (4.21) garante que cada posição (r, c) tenha no máximo um único contêiner. A restrição (4.22) é necessária para garantir que existam contêineres embaixo do contêiner que ocupa a célula (r, c) . A restrição (4.23) é responsável por definir como um contêiner pode ser desembarcado no porto d ao impor que se um contêiner que ocupa a posição (r, c) , então ele será desembarcado no porto d , se não houver um contêiner na posição $(r + 1, c)$ acima dele. A restrição (4.24) mantém a estabilidade do navio impondo um limite de altura. Este limite de altura é o teto da divisão do número de contêineres a bordo do navio pelo número de colunas. Desta forma, os contêineres são igualmente distribuídos no navio.

No Capítulo 6 é feita uma demonstração com um exemplo numérico que mostra a importância de se resolver esses dois problemas de forma conjunta e não separada. Na sequência, no Capítulo 7 são mostrados os resultados obtidos com o modelo CRP-MPSP proposto.

5 Justificativa

Este Capítulo tem como objetivo demonstrar através de exemplos que a figura do pátio é importante no problema do MPSP, e logo, o CRP e o MPSP devem ser resolvidos de forma integrada.

Duas hipóteses serão testadas, são elas:

- **Hipótese 1:** A organização dos contêineres no pátio pode alterar o plano de estiva para múltiplos portos.
- **Hipótese 2:** Existem casos em que resolver o CRP e o MPSP de forma integrada é melhor do que resolvê-los de forma sequencial.

Para comprovar as hipóteses 1 e 2, duas instâncias de teste foram criadas. A Tabela 5.1 descreve cada instância I em termos de: número de linhas do navio (R), número de colunas do navio (C), a quantidade de portos (P) percorridos pelo navio, a quantidade total de contêineres com os quais o navio lida ao longo de toda sua rota, e a quantidade total de variáveis e restrições que essa instância produz no modelo exato.

Vale a pena lembrar que o navio possui espaço suficiente para embarcar todos os contêineres durante sua rota.

Tabela 5.1 – Instâncias de Teste

I	Navio		Portos	Quantidade de Contêineres	Quantidade de Variáveis	Quantidade de Restrições
	R	C	P			
IT_A	5	4	10	29	265.641	110.906
IT_B	5	4	10	29	265.641	110.906

Note na Tabela 5.1 que as instâncias IT_A e IT_B são iguais. Além disso, ambas possuem a mesma matriz de transporte, como pode ser visto a seguir:

Esclarecidas as similaridades e diferenças das instâncias IT_A e IT_B , as hipóteses 1 e 2 podem ser testadas. Para isso, o modelo apresentado no Capítulo 4 foi implementado em Python e executado no Cplex 12.8 em um cluster com 8 nós, utilizando 20 threads e 120 GB de memória Ram.

Para a Hipótese 1, as instâncias IT_A e IT_B serão resolvidas pelo Cplex. Se o valor da função objetivo obtido para cada uma for diferente, então fica provado que a organização dos contêineres no pátio pode alterar o plano de estiva do navio.

Dessa forma, a Tabela 5.2 apresenta os resultados obtidos pelo *solver*. A coluna I mostra o número da instância testada, $F.O.$ traz o valor da função objetivo (equação 4.1), a coluna $Tempo$ traz o tempo em segundos o *solver* demorou em resolver a instância, a coluna $BestBound$ traz maior lower bound obtido pela árvore do *branch and bound*. Se o valor da função objetivo é igual ao $BestBound$, então a solução é ótima, e, por fim, coluna Gap traz o gap entre o valor do melhor objetivo inteiro e o objetivo do melhor nó restante.

Tabela 5.2 – Resultados das instâncias IT_A e IT_B .

I	$F.O.$	$Tempo$	$BestBound$	Gap
IT_A	3	19.816,38	3	0,00%
IT_B	10	2.797.755,76	5,2015	47,99%

Os resultados apresentados na Tabela 5.2 mostram que na instância IT_A , a otimização terminou em 19.816,38 segundos (5,50 horas), com 3 remanejamentos efetuados. Enquanto isso, na instância IT_B , depois de 32,38 dias, a otimização encontrou uma solução inteira com 10 remanejamentos e um gap de 47,99%. Apesar da otimização da instância IT_B não ter atingido o ótimo, como o *best bound* é 5,2015, isso significa que a solução ótima será algum valor inteiro entre 5,2015 e 10. Isto já é suficiente para mostrar que a instância IT_B nunca atingirá o mesmo valor que a instância IT_A .

Estes resultados são essenciais, uma vez que demonstram que apenas mudando a organização dos contêineres em um dos pátios da rota do navio, resultados bastante discrepantes foram atingidos pelo *solver*.

Além disso, também ficou comprovado que a organização nos pátios afeta a complexidade do problema, uma vez que o tempo da otimização da instância IT_A foi consideravelmente menor do que o tempo utilizado pelo *solver* na instância IT_B .

A Figura 5.3 mostra como ficou a organização dos contêineres no navio ao sair de cada porto p de sua rota na instância IT_A . Os contêineres em azul são aqueles que foram embarcados em cada porto p , e os contêineres em vermelho são aqueles que

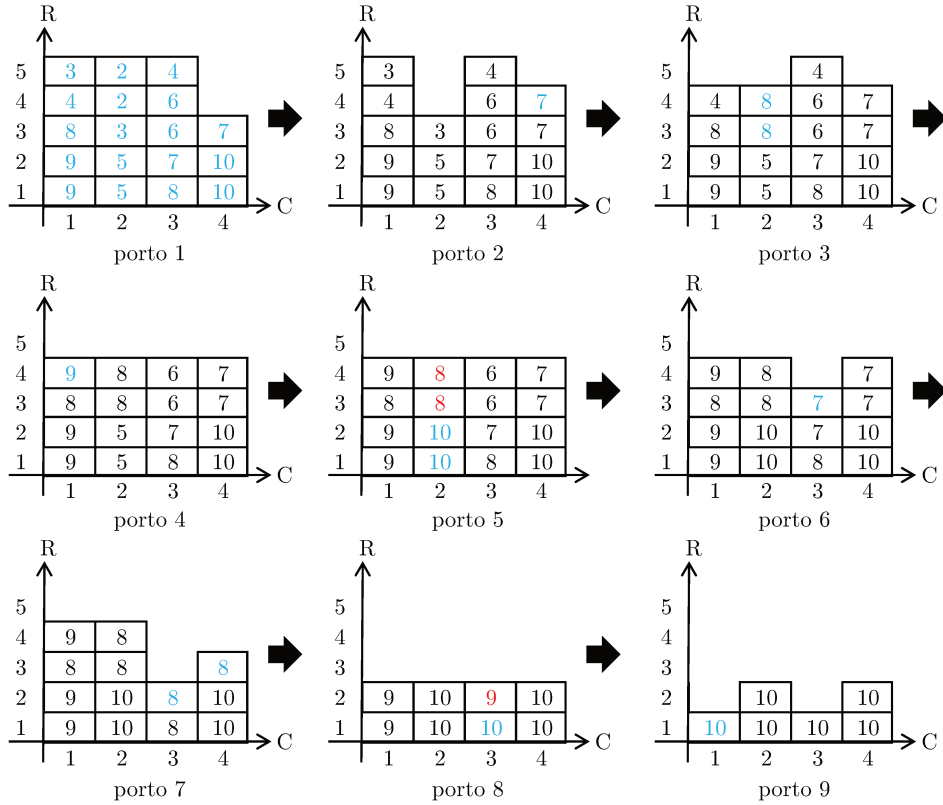


Figura 5.3 – Configuração do Navio em cada porto p da rota do navio na instância IT_A .

foram remanejados. Dado isso, observe que os três remanejamentos dessa instância foram efetuados no navio: dois contêineres com destino o oitavo porto foram remanejados no porto 5, e um contêiner com destino o nono porto foi remanejado no porto 8.

Para comprovar a Hipótese 2, deve-se mostrar que existem casos em que o valor da função objetivo do modelo integrado é melhor do que se os problemas fossem resolvidos separadamente.

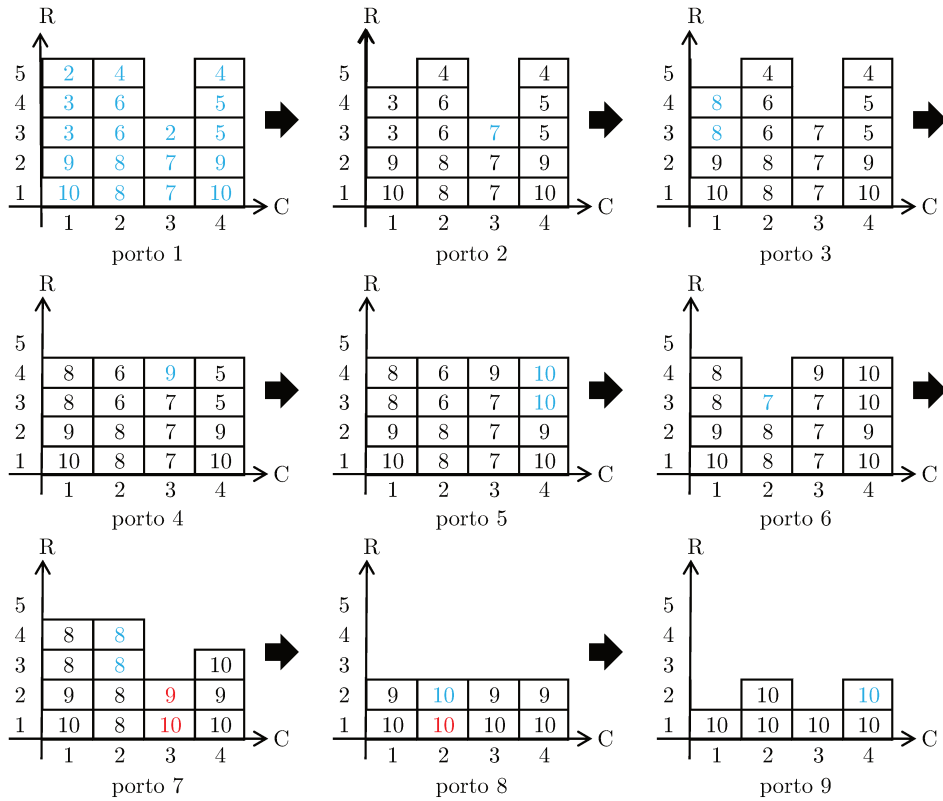
Para isso, no problema do CRP sem o MPSP, os contêineres possuem uma ordem de retirada pré-estabelecida. Supõem-se que essa ordem seja pré-estabelecida pela companhia marítima, através da resolução do MPSP. Em vista disso, primeiro deve-se resolver o MPSP isoladamente, e com a solução deste, obter a sequência de retirada dos contêineres do CRP para resolvê-lo.

O problema do MPSP não integrado é composto pela minimização da variável w_{odarc} , restrito pelas equações (4.20) até (4.24). Como a única diferença entre as instâncias IT_A e IT_B é o pátio do primeiro porto, no caso do MPSP sozinho não existe distinção entre cada uma destas instâncias, uma vez que os pátios não são considerados. Por este motivo, a solução do MPSP sem o CRP será chamada apenas de instância IT , e é apresentada na Tabela 5.3.

Os resultados apresentados na Tabela 5.3 mostram que na instância IT , a

Tabela 5.3 – Resultados da instância *IT*.

<i>I</i>	<i>F.O.</i>	<i>Tempo</i>	<i>BestBound</i>	<i>Gap</i>
<i>IT</i>	3	1,44	3	0,00%

Figura 5.4 – Configuração do Navio em cada porto p da rota do navio na instância *IT*.

otimização terminou em 1,44 segundos e com 3 remanejamentos efetuados. A Figura 5.4 mostra como ficou a organização dos contêineres no navio ao sair de cada porto p de sua rota na instância *IT*. Como feito anteriormente, os contêineres em azul são aqueles que foram embarcados em cada porto p , e os contêineres em vermelho são aqueles que foram remanejados. Dado isso, observe que dois contêineres, um com destino o nono porto e outro com destino o décimo porto foram remanejados no porto 7, e um contêiner com destino o décimo porto foi remanejado no porto 8.

Vale a pena ressaltar que a configuração do navio apresentada na Figura 5.4 foi obtida sem considerar a organização dos contêineres no pátio.

Por este motivo, vamos analisar com quantos remanejamentos seria possível retirar todos os contêineres do pátio do primeiro porto da instância *IT_B* e obter o navio na configuração do porto 1 mostrado na Figura 5.4.

Para facilitar, as Figuras 5.5 e 5.6 ilustram o raciocínio a ser explanado. Nestas

figuras, os contêineres do pátio do primeiro porto da instância IT_B estão representados pelo seu porto de destino, e não pelo seu índice.

Assim sendo, veja que na base do navio na configuração do porto 1 mostrado na Figura 5.4, temos contêineres com destino aos portos 10, 8 e 7. Ou seja, algum desses contêineres deve ser retirado primeiro do pátio. Se escolhermos o contêiner 7 para ser retirado do pátio primeiro, 5 realocações já seriam geradas. Esta seria a melhor escolha, visto que os contêineres com destino aos portos 8 ou 10 gerariam 6 e 9 realocações, respectivamente. Em seguida, seria possível retirar os contêineres 8 sem nenhuma realocação. Estas operações podem ser visualizadas na Figura 5.5.

Na sequência, dos contêineres que deveriam ser retirados, o que gera o menor número de realocações é o contêiner com destino ao porto 10 e se encontra na posição (1,1) do pátio. Para que esse contêiner seja retirado, é necessário mover o contêiner 9 da posição (2,1) para a posição (6,3). Com isso, o contêiner 9 que agora se encontra no topo da terceira coluna do pátio também pode ser retirado. Estas operações podem ser visualizadas na Figura 5.6.

Neste momento, o número total de realocações no pátio são 6, entretanto, pelo menos mais 6 realocações seriam necessárias para se retirar todos os contêineres deste pátio (veja a Figura 5.7) e obter o navio na configuração do porto 1 como mostrado na Figura 5.4.

Desse modo, o problema resolvido de forma sequencial traria 3 realocações no navio e 12 realocações no pátio do primeiro porto, resultando em um total de 15 realocações. Enquanto isso, o problema resolvido de forma integrada, como proposto no modelo CRP-MPSP, obtém uma solução com pelo menos 10 realocações em toda a rota do navio.

Com estes resultados, fica claro que o CRP e o MPSP devem ser resolvidos de forma integrada, uma vez que a configuração do pátio pode ter impacto direto no navio. Demonstrada a necessidade de integração do CRP e do MPSP, o Capítulo seguinte apresenta os testes computacionais do modelo integrado e faz a comparação de dois *solvers* comerciais diferentes, o Cplex 12.8 e o Gurobi 8.0.

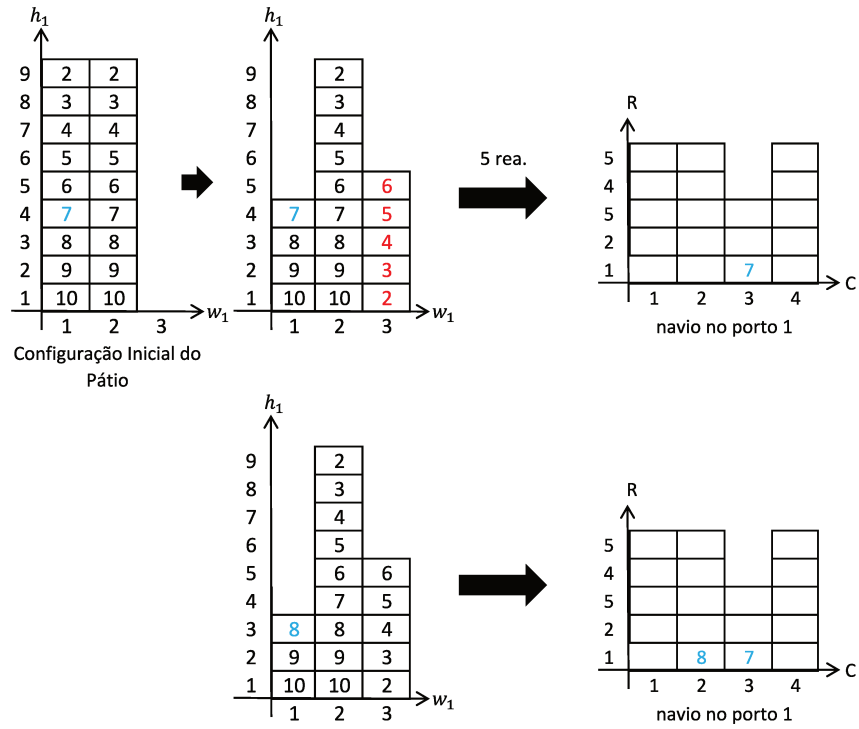


Figura 5.5 – Sequência de retirada dos contêineres do pátio do primeiro porto da instância IT_B , dado o plano de estiva apresentado na Figura 5.4 - Parte 1.

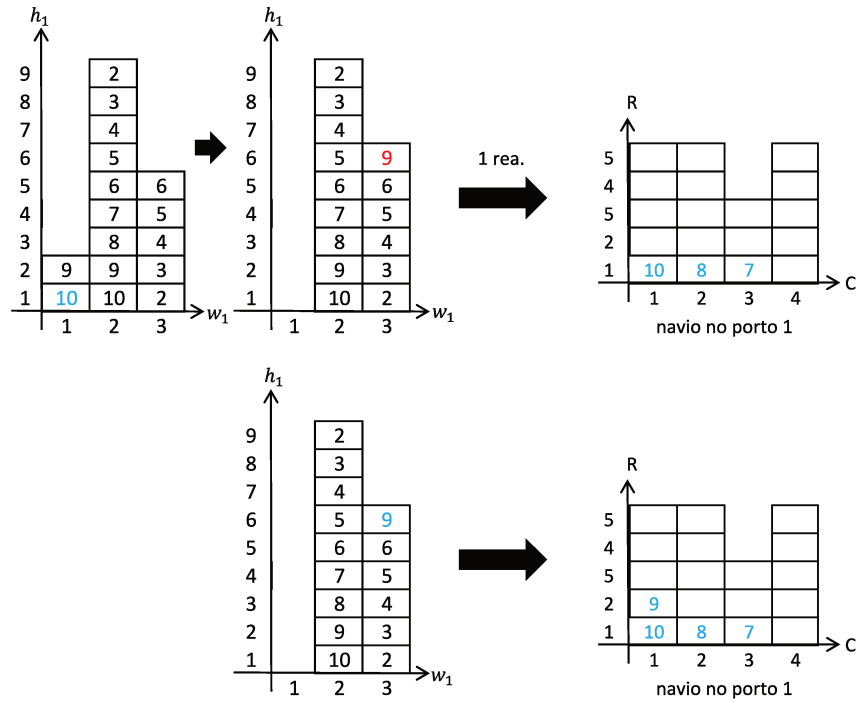


Figura 5.6 – Sequência de retirada dos contêineres do pátio do primeiro porto da instância IT_B , dado o plano de estiva apresentado na Figura 5.4 - Parte 2.

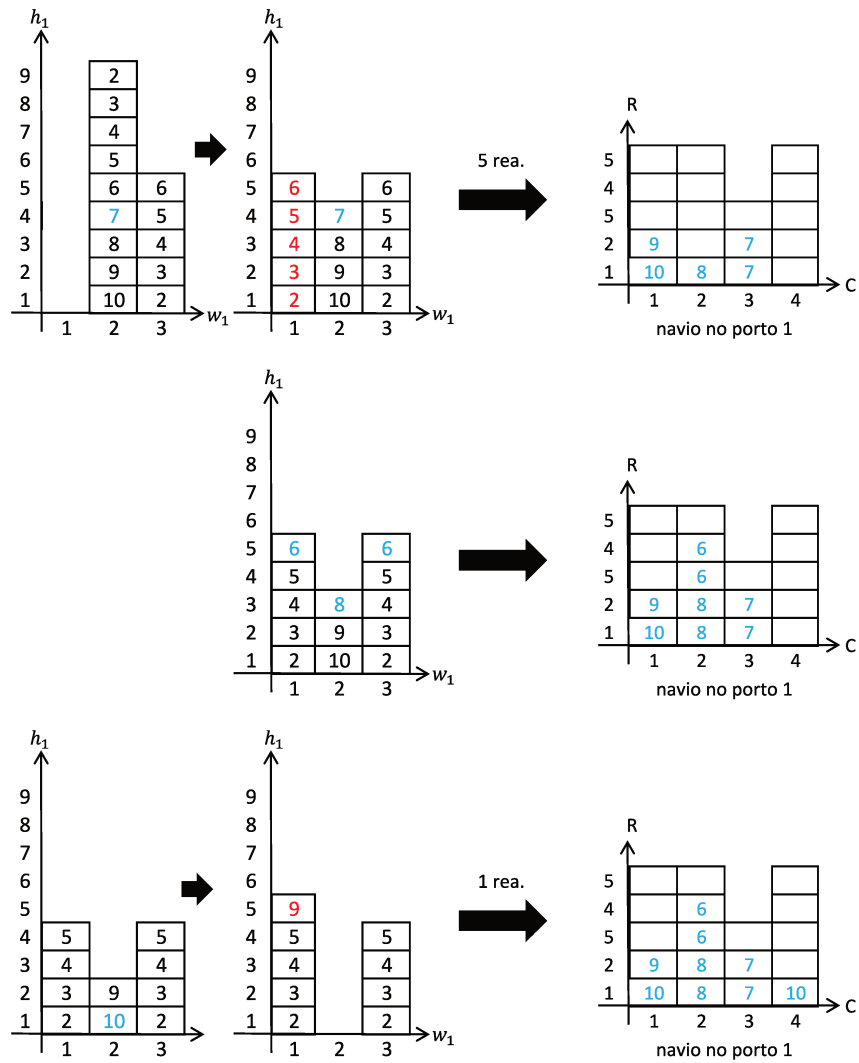


Figura 5.7 – Sequência de retirada dos contêineres do pátio do primeiro porto da instância IT_B , dado o plano de estiva apresentado na Figura 5.4 - Parte 3.

6 Técnica de Solução

A técnica de solução proposta consiste em gerar soluções iniciais factíveis para o CRP-MPSP e inseri-las no *solver* de otimização junto com o modelo. Este procedimento é conhecido como *warm-start*.

A vantagem de se usar *warm-start* é que ele permite que o *solver* elimine parcelas do espaço de busca e, portanto, pode resultar em árvores (*branch-and-cut*) menores, possivelmente levando a uma solução ótima mais rapidamente.

Para gerar as soluções iniciais factíveis para o CRP-MPSP rapidamente, duas regras heurísticas simples foram desenvolvidas, chamadas de HR_1 e HR_2 . Ambas resolvem o problema no seu formato matricial, e não binário.

Esta abordagem foi escolhida uma vez que o problema claramente não possui uma estrutura decomponível, e técnicas de decomposição, como Benders, não se mostraram eficientes.

Cada uma das regras heurísticas será explicada em detalhes a seguir. Na sequência, é explicado como a solução obtida é utilizada como *warm-start*.

6.1 Regras Heurísticas

As regras heurísticas HR_1 e HR_2 são inspiradas na heurística proposta por (CASERTA *et al.*, 2012) e foram devidamente adaptadas para resolver o problema integrado. Em (CASERTA *et al.*, 2012), a regra heurística proposta retira os contêineres do pátio seguindo uma ordem pré-definida. Caso um contêiner precise ser remanejado, uma posição é escolhida verificando em cada coluna do pátio, se existe uma posição na qual este contêiner poderia ser alocado sem gerar um remanejamento futuro, ou seja, se já não existe um contêiner nesta pilha que deverá ser retirado antes do contêiner que está sendo remanejado. Se não existir uma posição com essa característica, a regra busca a posição que gere o remanejamento mais tardio.

No CRP-MPSP, como não há uma ordem de retirada pré-definida, a regra HR_1 , e também a regra HR_2 , são programadas para retirar do pátio primeiro os contêineres que possuem o porto de destino mais distante. Dessa forma, estes contêineres podem ser alocados na base do navio. Adicionalmente, para evitar que remanejamentos sejam efetuados nos pátios, as regras olham apenas os contêineres que estão no topo de cada coluna do pátio. Isto é, dentre os contêineres que estão no topo de cada coluna do pátio,

o que será escolhido para ser retirado primeiro é aquele com o porto de destino mais distante.

Na sequência, o contêiner retirado do pátio deve ser carregado no navio. Para isso, as regras devem escolher uma posição dentre todas as posições disponíveis, e é nesta etapa em que a regra HR_1 se diferencia da regra HR_2 .

No caso da regra HR_1 , o procedimento consiste em olhar para cada coluna do navio que possui uma posição disponível e verificar se existe uma posição na qual o contêiner que está sendo carregado pode ser alocado sem que tenha que ser remanejado em um porto antes do seu destino final e sem que o limite de altura máximo seja atingido. Caso não exista nenhuma posição com essa característica, a posição escolhida é aquela que gera o remanejamento no porto mais distante possível.

Enquanto isso, a regra HR_2 também verifica se existe uma posição na qual o contêiner que está sendo carregado possa ser alocado sem que tenha que ser remanejado em um porto antes do seu destino final e sem que o limite de altura máxima seja atingido, mas, se não houver nenhuma posição com essa característica, a regra HR_2 aleatoriamente escolhe uma posição disponível para alocar o contêiner que está sendo carregado no navio.

Para o descarregamento, ambas as regras seguem o mesmo procedimento, que consiste em verificar quais contêineres tem destino o porto p no qual o navio se encontra e descarregá-los. Se algum remanejamento precise ser efetuado, esses contêineres de bloqueio também são descarregados e alocados em uma área separada para estes contêineres. Depois que o descarregamento é finalizado, estes contêineres são carregados novamente no navio, utilizando mesmo procedimento de carregamento descrito anteriormente.

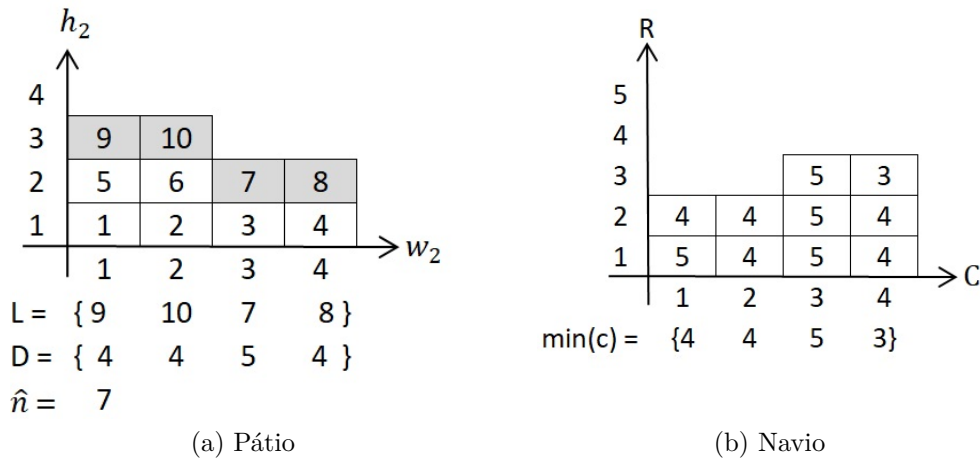
Estes procedimentos se repetem em todos os portos pertencentes à rota do navio. Lembrando que no último porto, apenas o descarregamento do navio é efetuado.

A seguir é feita uma descrição mais formal das regras HR_1 e HR_2 .

6.1.1 Regra Heurística HR_1

Dada uma instância I , seja L o conjunto dos contêineres atualmente localizados no topo de cada coluna do $Patio(o)$, sendo $o = \{1, \dots, P - 1\}$; ou seja, L é o conjunto de contêineres que podem ser removidos do pátio. Na Figura 6.1 (a), por exemplo, tem-se que $L = \{9, 10, 7, 8\}$.

Para determinar qual dos contêineres em L deve ser retirado primeiro, é verificado qual deles tem o porto de destino mais distante. Essa informação está disponível no parâmetro $\phi_{o,d}$. Logo, supondo que os respectivos portos de destino dos contêineres contidos no conjunto L seja $D = \{4, 4, 5, 4\}$, a heurística escolherá o contêiner $\hat{n} = 7$ para

Figura 6.1 – Primeiros passos da Regra Heurística HR_1

ser retirado primeiro e carregado no navio, uma vez que o seu destino é o mais distante. No caso de um empate nesta etapa, a heurística escolhe o primeiro contêiner da lista que tem o destino mais distante.

Note que, uma vez que a regra remove apenas os contêineres que estão no topo de suas colunas, não são feitos remanejamentos no pátio.

Para decidir em qual posição do navio este contêiner deve ser colocado, é calculada uma pontuação para cada coluna, baseado no porto de destino mais próximo dos contêineres que já estão a bordo do navio. Para isso, define-se $\min(c)$ como o porto de destino mais próximo do contêiner na coluna c , sendo $c = \{1, \dots, C\}$. No exemplo da Figura 6.1 (b), esta pontuação seria $\min(c) = \{4, 4, 5, 3\}$. No caso de uma coluna vazia, computa-se $\min(c) = P + 1$, e no caso de uma coluna cheia, $\min(c) = 0$.

Se houver uma coluna onde $\min(c)$ é maior ou igual que o porto de destino do contêiner \hat{n} , então deve-se checar se essa posição respeita o limite de altura imposto neste porto pela restrição (4.24). Em caso positivo, a coluna escolhida é a coluna diferente de zero que minimiza $\min(c)$. O objetivo neste passo é não colocar o contêiner \hat{n} em cima de outro contêiner que será retirado antes do navio, evitando remanejamentos adicionais nos portos seguintes.

Se não houver uma coluna que satisfaça essa condição, então escolher a coluna onde $\min(c)$ é maximizado, pois o bloco \hat{n} causará um novo remanejamento forçado de qualquer maneira. A ideia com esta escolha é postergar ao máximo o remanejamento que está sendo gerado.

Brevemente, a escolha da coluna c^* na qual o contêiner \hat{n} deve ser alocado é feita da seguinte forma:

$$c^* = \begin{cases} \operatorname{argmin}_{c \in \{1, \dots, C\}} \{ \min(c) : \min(c) \geq \hat{n} \}, & \text{se } \exists c : \min(c) \geq \hat{n}, \\ \operatorname{argmax}_{c \in \{1, \dots, C\}} \{ \min(c) \}, & \text{caso contrário.} \end{cases} \quad (6.1)$$

No caso do exemplo utilizado, o contêiner $\hat{n} = 7$ é alocado na coluna $c^* = 3$, cujo $\operatorname{argmin}_{c \in \{1, \dots, C\}} \{ \min(c) : \min(c) \geq \hat{d} \} = 5$, como ilustrado na Figura 6.2.

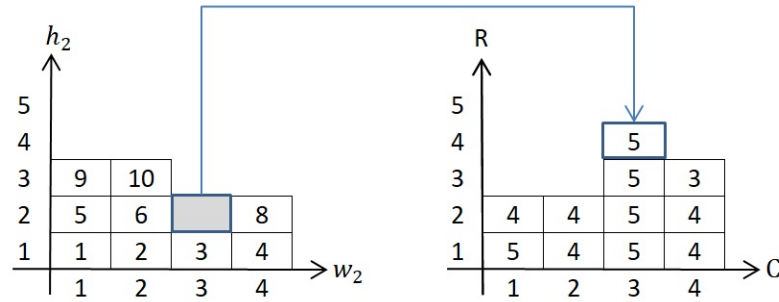


Figura 6.2 – Regra Heurística HR_1 : Alocando o contêiner \hat{n} na coluna c^* do navio.

Se depois de carregar todos os contêineres do porto p ainda houver um contêiner acima da altura máxima permitida, uma nova posição será escolhida para ele, seguindo as etapas descritas anteriormente. Terminado o carregamento no porto p , o navio segue para o porto seguinte $p + 1$.

Vale a pena lembrar que do segundo até o porto $P - 1$, primeiro é efetuado o descarregamento dos contêineres com destino o porto p e depois o carregamento dos contêineres que estão no pátio deste porto e tem como destino os portos seguintes.

Logo, para descarregar, a regra heurística identifica onde estão os contêineres que devem ser descarregados no porto p no qual o navio se encontra. Este conjunto é chamado de *Unload*. Em seguida é verificado, um a um, se estes contêineres se encontram no topo de sua coluna. Se estiver no topo, então este contêiner é descarregado. Caso contrário, o conjunto *Rel* é definido. Este conjunto contém os contêineres que estão bloqueando a saída do contêiner objetivo, de acordo com a política LIFO. Como remanejamentos não são permitidos dentro do navio, esses contêineres são descarregados para uma área reservada no pátio, chamada de *Temp_Yard*, e depois carregados novamente após todo o descarregamento ter sido feito.

Terminado o descarregamento é repetido o procedimento de carregamento descrito acima para o pátio do porto atual e também para a área reservada onde foram alocados os contêineres que foram remanejados.

Ao chegar no último porto, apenas o procedimento de descarregamento descrito é repetido.

O Algoritmo 6.1 apresenta a regra heurística HR_1 como um pseudocódigo.

Algoritmo 6.1 Regra Heurística

```

1: initialize(instância do problema)
2:  $p \leftarrow 1$ ,  $move\_counter \leftarrow 0$ 
3: for  $p = \{1, \dots, P\}$  do
4:   if ( $p = 1$ ) then
5:     while  $Yard(p) \neq \emptyset$  do
6:       Defina o conjunto  $L$  e o conjunto  $D$ 
7:        $\hat{n} = \max(D)$ 
8:       for  $c = \{1, \dots, C\}$  do
9:         if Coluna  $c$  está vazia then
10:           $\min(c) = P + 1$ 
11:        end if
12:        if Coluna  $c$  está cheia then
13:           $\min(c) = 0$ 
14:        end if
15:        if Coluna  $c$  não está vazia e nem cheia then
16:           $\min(c) = \min\{k : \text{porto de destino } k \text{ na coluna } c\}$ 
17:        end if
18:      end for
19:      Encontre  $c^*$  usando Eq. (6.1)
20:       $Patio(p) \leftarrow Patio(p) \setminus \{\hat{n}\}$ 
21:      Coloque  $\hat{n}$  na coluna  $c^*$ 
22:    end while
23:  end if
24:   $p \leftarrow p + 1$ 
25:  if ( $1 < p < P$ ) then
26:    Defina o conjunto  $Unload$  {conjunto de contêineres a serem descarregados em  $p$ } e o conjunto
     $Temp\_Yard$  {Pátio para alocar os contêineres de remanejamento}
27:    while  $Unload \neq \emptyset$  do
28:      Estabeleça  $\hat{n}$  como o primeiro elemento em  $Unload$ 
29:      if  $\hat{n}$  está no topo de sua coluna then
30:         $Unload \leftarrow Unload \setminus \{\hat{n}\}$ 
31:      else
32:        Defina o conjunto  $Rel$ 
33:        Estabeleça  $\hat{r}$  como o primeiro elemento em  $Rel$ 
34:        Mova  $\hat{r}$  para  $Temp\_Yard$  utilizando o mesmo procedimento descrito quando  $p = 1$ 
35:         $Unload \leftarrow Unload \setminus \{\hat{r}\}$ 
36:         $move\_counter \leftarrow move\_counter + 1$ 
37:      end if
38:    end while
39:    Mova os contêineres de  $Temp\_Yard$  de volta para o navio utilizando o mesmo procedimento
    descrito quando  $p = 1$ 
40:    Repita o passo de carregamento como descrito quando  $p = 1$ 
41:    if Eq. (4.24) é violada then
42:      Encontre o novo  $c^*$  utilizando a Eq. (6.1)
43:       $move\_counter \leftarrow move\_counter + 1$ 
44:    end if
45:     $p \leftarrow p + 1$ 
46:  end if
47:  if ( $p = P$ ) then
48:    Execute somente o procedimento de descarregamento
49:  end if
50: end for
  
```

6.1.2 Regra Heurística HR_2

Como dito anteriormente, o que diferencia a regra heurística HR_2 da regra heurística HR_1 é apenas a forma de escolher uma posição para um contêiner que está sendo carregado no navio quando não existem posições que não gerarão remanejamentos futuros. Enquanto HR_1 utiliza a função (6.1) que visa postergar ao máximo o remanejamento futuro, HR_2 aleatoriamente escolhe uma posição disponível.

Dessa forma, a escolha da coluna c^* na qual o contêiner \hat{n} deve ser alocado é feita da seguinte forma por HR_2 :

$$c^* = \begin{cases} \operatorname{argmin}_{c \in \{1, \dots, C\}} \{ \min(c) : \min(c) \geq \hat{n} \}, & \text{se } \exists c : \min(c) \geq \hat{n}, \\ \operatorname{datasample}(\min(c), 1), & \text{caso contrário.} \end{cases} \quad (6.2)$$

A função $\operatorname{datasample}(\min(c), 1)$ retorna um valor amostrado uniformemente de forma aleatória em $\min(c)$.

No Algoritmo (6.1), a modificação seria na linha 19, que seria substituída por: “Encontre c^* usando Eq. (6.2)”.

O resultado da regra HR_1 é fixado como limitante superior para a regra HR_2 . Para garantir que em algum momento HR_2 obtenha uma solução melhor do que HR_1 , a regra HR_2 é executada 10.000 vezes. Se uma solução melhor aparecer, então esta solução é mantida até que uma melhor seja obtida.

A motivação por trás da modificação proposta pela heurística HR_2 se dá pela hipótese de que quanto menor o *gap* entre a solução inicial gerada pela heurística e o valor da solução ótima, maior é o impacto do *warm-start* e, conseqüentemente, no tempo computacional necessário para que a solução ótima seja encontrada.

6.2 Warm-Start

Para utilizar o resultado das regras heurísticas como *warm-start*, é preciso transformar a solução do formato matricial para o formato binário. Isto é, é preciso dar valor às variáveis do problema, que foram apresentados na seção 4.1.1.

Vale a pena ressaltar que as regras heurísticas HR_1 e HR_2 sempre geram soluções factíveis, e portanto, podem ser utilizadas como *warm-start* no modelo CRP-MPSP.

Para ilustrar este procedimento, um exemplo será apresentado.

Suponha que a solução de uma certa instância, por qualquer uma das regras heurísticas, resultou em uma sequência de retirada dos contêineres de um pátio como a apresentada na Figura 6.3, e que este é o primeiro pátio da rota do navio. Suponha também que a ocupação do navio dessa mesma instância resultou ser como a apresentada na Figura 6.4.

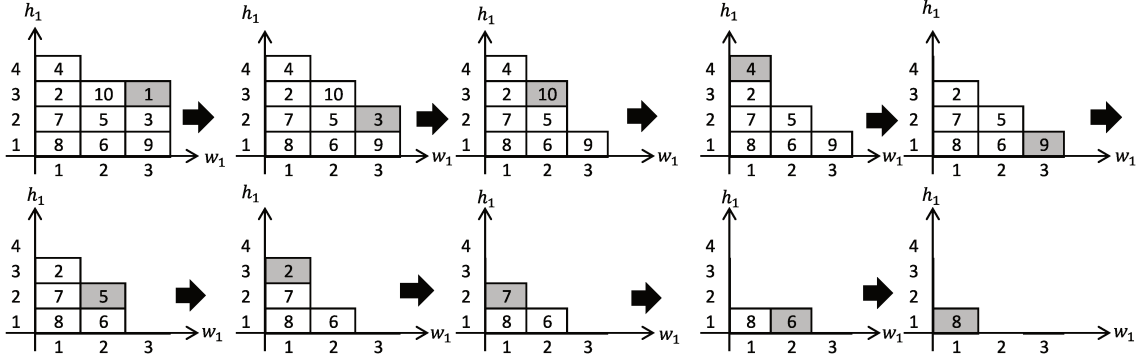


Figura 6.3 – Sequência de retirada dos contêineres de um pátio portuário.

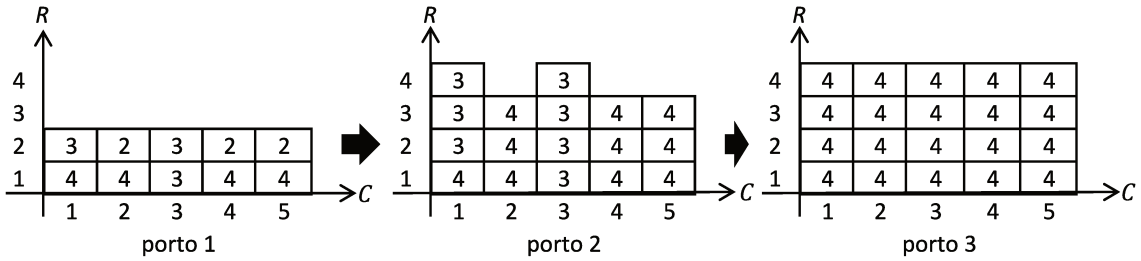


Figura 6.4 – Representação da ocupação de navio em cada porto p de uma rota com três portos.

É importante estar atento de que no pátio, uma posição é uma coordenada (i, j) , onde $i \in \{W_o\}$ representa as colunas do pátio e $j \in \{H_o\}$ representa as linhas do pátio, sendo que $o \in \{P - 1\}$.

Com isso, observe na Figura 6.3 que a sequência de retirada dos contêineres do pátio é $\{1 - 3 - 10 - 4 - 9 - 5 - 2 - 7 - 6 - 8\}$; logo, para cada variável:

- A variável b_{ijnt} indica a posição (i, j) do contêiner n no período de tempo t . Assim, pode-se dizer que
 - O contêiner 1 é o primeiro a ser retirado do pátio, então tem-se que $b_{3,3,1,1} = 1$, o que significa que o contêiner 1 está na posição (3, 3) no tempo 1, e todos os seguintes $b_{i,j,1,t}$ devem ser zero.
 - O contêiner 3 é o segundo a ser retirado do pátio, então tem-se que $b_{3,2,3,1} = 1$ e $b_{3,2,3,2} = 1$, o que significa que o contêiner 3 esteve na posição (3, 2) no tempo

- 1 e no tempo 2, quando foi retirado. Todos os outros $b_{i,j,3,t}$ devem ser zero. O mesmo raciocínio se repete até que todos os contêineres sejam retirados do pátio.
- O último contêiner a ser retirado deste pátio é o contêiner 8, no tempo 10. Logo, $b_{1,1,8,t} = 1$ para $\forall t \in \{T_1\}$, e $b_{i,j,8,t} = 0$, $\forall i \in \{W_1\} \setminus \{i = 1\}$, $j \in \{H_1\} \setminus \{j = 1\}$ e $t \in \{T_1\}$.
 - Repetir o procedimento para todos os $P - 1$ pátios.
- A variável y_{ijnt} indica se o contêiner n é retirado de (i, j) no período de tempo t . Assim, em relação ao exemplo apresentada na Figura 6.3, pode-se dizer que
 - Como o contêiner 1 é o primeiro a ser retirado do pátio, tem-se que $y_{3,3,1,1} = 1$ e $y_{i,j,1,t} = 0$; $\forall i \in \{W_1\} \setminus \{i = 3\}$, $j \in \{H_1\} \setminus \{j = 3\}$ e $t = \{2, \dots, T_1\}$.
 - Como o contêiner 3 é o segundo a ser retirado do pátio, tem-se que $y_{3,2,3,1} = 1$ e $y_{3,2,3,2} = 1$. Todos os outros $y_{i,j,3,t}$ devem ser zero.
 - O último contêiner a ser retirado deste pátio é o contêiner 8, no tempo 10. Logo, $y_{1,1,8,t} = 1$ para $\forall t \in \{T_1\}$, e $y_{i,j,8,t} = 0$, $\forall i \in \{W_1\} \setminus \{i = 1\}$, $j \in \{H_1\} \setminus \{j = 1\}$ e $t \in \{T_1\}$.
 - Repetir o procedimento para todos os $P - 1$ pátios.
 - A variável v_{nt} indica se o contêiner n foi retirado no período t' , onde $t' \in \{1, \dots, t - 1\}$,
 - Como o contêiner 1 é o primeiro a ser retirado do pátio, neste caso teríamos que $v_{1,1} = 0$, e $v_{1,t} = 1$ para $t = \{2, \dots, T_1\}$.
 - Como o contêiner 3 é o segundo a ser retirado do pátio, teríamos que $v_{3,1} = 0$ e $v_{3,2} = 0$; e $v_{3,t} = 1$ para $t = \{3, \dots, T_1\}$.
 - O último contêiner a ser retirado deste pátio é o contêiner 8, no tempo 10. Logo, $v_{8,t} = 0$, $\forall t \in \{T_1\}$.
 - Repetir o procedimento para todos os $P - 1$ pátios.
 - A variável x_{ijklnt} indica, no pátio, se o contêiner n é realocado de (i, j) para (k, l) no período de tempo t . No caso do exemplo apresentado, como não houveram remanejamentos no pátio, então tem-se que $x_{ijklnt} = 0$; $\forall (i, k) \in \{W_o\}$, $(j, l) \in \{H_o\}$, $n \in \{N_o\}$ e $t \in \{T_1\}$.
 - A variável z_{ntrc} indica se no período de tempo t , o contêiner n ocupa a posição (r, c) no navio. Esta variável pode ser construída através da observação da sequência de carregamento do navio, como ilustrado na Figura 6.5.

- O contêiner 1, que foi o primeiro contêiner a ser carregado no navio, possui como destino o porto 4 e foi carregado na posição (1,4), durante o tempo $t = 1$. Logo, tem-se que $z_{1,t,1,4} = 1$ para $t \in \{T_1\}$, e os demais $z_{1,t,r,c} = 0$, $\forall t \in \{T_1\}, r \in \{R\} \setminus \{r = 4\}$ e $c \in \{C\} \setminus \{c = 4\}$.
- O contêiner 3, que foi o segundo contêiner a ser carregado no navio, possui como destino o porto 2 e foi carregado na posição (2,4), durante o tempo $t = 2$. Logo, tem-se que $z_{3,t,2,4} = 1$ para $t = \{2, \dots, T_1\}$, e os demais $z_{3,t,r,c} = 0$.
- O contêiner 10, que foi o terceiro contêiner a ser carregado no navio, possui como destino o porto 4 e foi carregado na posição (1,2), durante o tempo $t = 3$. Logo, tem-se que $z_{10,t,1,2} = 1$ para $t = \{3, \dots, T_1\}$, e os demais $z_{10,t,r,c} = 0$.
- Repetir o procedimento para todos os N_o contêineres, onde $o = \{1, \dots, P - 1\}$.

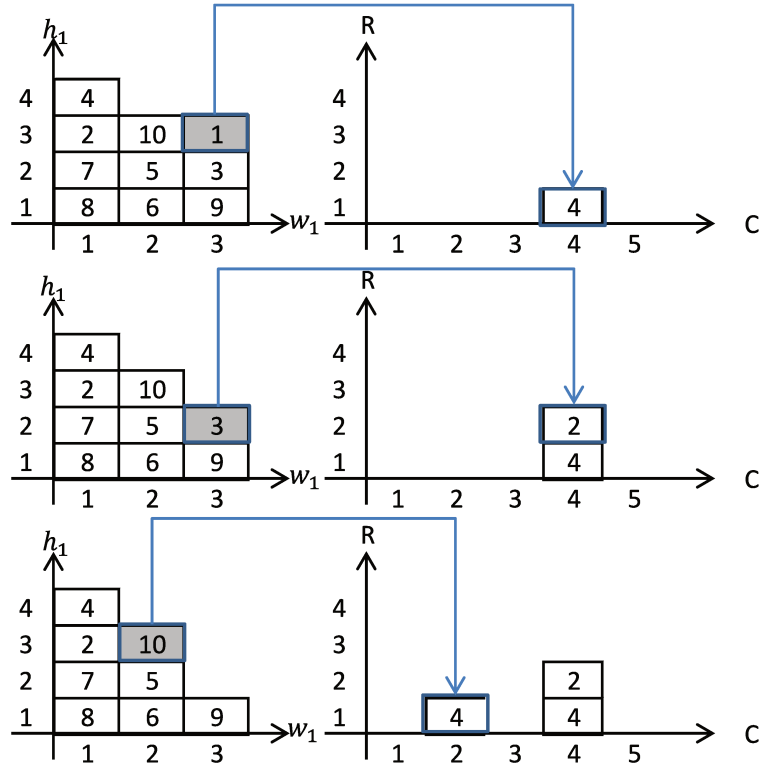


Figura 6.5 – Representação da retirada de contêineres do pátio e carregamento no navio.

- A variável u_{orc} indica se ao sair do porto o , a posição (r, c) do navio está ocupada por um contêiner. Para construir essa variável, basta observar a configuração final do navio depois de terminar o carregamento em cada porto $o = \{1, \dots, P - 1\}$, como ilustrado na Figura 6.4.
- Ao sair do primeiro porto, as linhas 1 e 2 do navio estão completamente ocupadas. Com essa informação tem-se que $u_{1,r,c} = 1$ quando $r = \{1, 2\}$ e $\forall c \in \{C\}$, e $u_{1,r,c} = 0$ quando $r = \{3, 4\}$ e $\forall c \in \{C\}$.

- Ao sair do segundo porto, as únicas posições vazias no navio são: $(4,2)$, $(4,4)$ e $(4,5)$. Logo, $u_{2,4,2} = 0$, $u_{2,4,4} = 0$ e $u_{2,4,5} = 0$, os demais $u_{2,r,c}$ devem ser iguais a 1.
- Ao sair do terceiro porto, não há nenhuma posição vazia no navio, então todos os $u_{3,r,c}$ devem ser iguais a 1, onde $r \in \{R\}$ e $c \in \{C\}$.
- A variável q_{odrc} indica se um contêiner com destino final o porto d é remanejado no porto o e realocado na posição (r, c) no navio. No exemplo proposto não foram efetuados remanejamentos no navio, então $q_{odrc} = 0$, $\forall o = \{1, \dots, P-1\}, d = \{o+1, \dots, P\}, r \in \{R\}, c \in \{C\}$.
- A variável w_{odarc} indica se há um contêiner na posição (r, c) do navio que foi embarcado no porto o , com destino final no porto d e remanejado no porto a . Novamente, como no exemplo proposto não foram efetuados remanejamentos no navio, então $w_{odarc} = 0$, $\forall o = \{1, \dots, P-1\}, d = \{o+1, \dots, P\}, a = \{o+1, \dots, d-1\}, r \in \{R\}, c \in \{C\}$.

Cada variável do modelo é inserida no *solver* com seu respectivo valor, como mostrado no processo descrito nesta seção. Esses valores não são fixos, dessa forma o modelo pode seguir a otimização em busca de soluções melhores, partindo da solução inserida.

No próximo capítulo serão apresentados os resultados computacionais obtidos com o modelo exato, com as regras heurísticas HR_1 e HR_2 , e com o modelo com *warm-start*.

7 Resultados Computacionais

Neste capítulo, primeiro serão apresentados os resultados computacionais obtidos com o modelo exato, com as regras heurísticas HR_1 e HR_2 , e com o modelo com *warm-start*.

7.1 Resultados do CRP-MPSP

O modelo CRP-MPSP, apresentado no Capítulo 4, foi implementado em Python e executado em um computador com AMD Ryzen™ 7 1700, 3 GHz, 64 GB RAM.

As Tabelas 7.1 e 7.2 descrevem as instâncias de teste geradas aleatoriamente (com distribuição uniforme). Para cada instância I estão descritos: número de linhas do pátio (H), número de colunas do pátio (W), número de linhas do navio (R), número de colunas do navio (C), a quantidade de portos (P) percorridos pelo navio, a quantidade total de contêineres com os quais o navio lida ao longo de toda sua rota, e por fim, a quantidade total de variáveis e restrições que essa instância produz no modelo exato.

Os pátios de todas as instâncias possuem ocupação de 65% a 85%. Para cada instância da Tabela 7.1, o navio criado possui somente o espaço mínimo necessário para embarcar todos os contêineres a serem entregues ao longo de sua rota.

Note que as instâncias apresentadas na Tabela 7.2 são as mesmas instâncias da Tabela 7.1, mas com o número de colunas do navio dobradas. Com isso, podemos analisar o impacto de se ter mais espaço para manusear os contêineres no navio nos resultados do problema. Note também que aumentar o número de colunas do navio aumenta o tamanho do problema em termos de quantidade de variáveis e restrições.

As instâncias 12A e 12B aparecem com um “*” nas colunas referentes à quantidade de variáveis e restrições, porque a memória disponível não foi suficiente para terminar de construir os modelos.

As instâncias de teste apresentadas nas Tabelas 7.1 e 7.2 foram resolvidas em dois *solvers* comerciais diferentes: o Cplex 12.8 e o Gurobi 8.0. A motivação para a utilização desses dois solvers é comparar os resultados e verificar se alguns deles se mostram mais eficientes para o CRP-MPSP.

Dado isso, as Tabelas 7.3 e 7.4 reportam os resultados computacionais obtidos pelo Cplex e pelo Gurobi para o modelo integrado. Ao longo do experimento foi fixado um tempo computacional máximo de 432.000 segundos (5 dias), que ao ser atingido o *solver*

Tabela 7.1 – Instâncias de teste do grupo *A*

<i>I</i>	Pátio		Navio		Portos	Quantidade de Contêineres	Quantidade de Variáveis	Quantidade de Restrições
	<i>H</i>	<i>W</i>	<i>R</i>	<i>C</i>	<i>P</i>			
1A	4	3	4	5	4	30	57.080	30.373
2A	4	3	4	6	6	46	83.924	45.864
3A	4	3	5	9	10	92	213.155	123.772
4A	4	3	6	10	15	137	351.493	204.102
5A	4	6	5	5	4	58	757.955	223.701
6A	4	6	6	6	6	91	1.125.877	338.455
7A	4	6	7	9	10	162	2.053.070	665.730
8A	4	6	7	10	15	261	3.773.715	1.407.996
9A	6	10	6	14	4	131	21.861.321	3.340.081
10A	6	10	6	19	6	212	34,582,630	5,531,912
11A	6	10	6	30	10	409	73,054,437	12,664,333
12A	6	10	6	44	15	617	*	*

Tabela 7.2 – Instâncias de teste do grupo *B*

<i>I</i>	Pátio		Navio		Portos	Quantidade de Contêineres	Quantidade de Variáveis	Quantidade de Restrições
	<i>H</i>	<i>W</i>	<i>R</i>	<i>C</i>	<i>P</i>			
1B	4	3	4	10	4	30	63.460	37.095
2B	4	3	4	12	6	46	95.516	57.627
3B	4	3	4	18	10	92	265.760	172.915
4B	4	3	4	20	15	137	473.653	300.916
5B	4	6	5	18	4	58	809.660	277.261
6B	4	6	6	16	6	91	1.208.629	423.569
7B	4	6	7	20	10	162	2.273.640	885.553
8B	4	6	7	36	15	261	4.478.055	2.072.990
9B	6	10	6	28	4	131	22.345.497	3.833.009
10B	6	10	6	38	6	212	35.616.724	6.578.547
11B	6	10	6	60	10	409	76.462.917	16.099.967
12B	6	10	6	88	15	617	*	*

é parado e o resultado atingido até então é reportado. A coluna *I* mostra o número da instância testada (descritas nas Tabelas 7.1 e 7.2), *F.O.* traz o valor da função objetivo (Equação 4.1), a coluna *Tempo* traz o tempo em segundos que cada *solver* demorou em resolver a instância, e, por fim, coluna *Gap* traz o *gap* da solução encontrada em relação ao *Best Bound*.

Repare que algumas instâncias possuem um “—” nas colunas que deveriam reportar o valor da função objetivo (F.O.) e o gap. Este símbolo representa instâncias que atingiram o tempo máximo permitido e nenhuma solução factível foi encontrada.

As últimas quatro instâncias, tanto da Tabela 7.3, quanto da Tabela 7.4, estão preenchidas com um “*”. Isso significa que os 64GB de memória RAM disponíveis não foram suficientes para o *solver* resolver determinada instância.

Dentre as instâncias com problemas de memória, dois tipos de erro ocorrem.

Tabela 7.3 – Resultados Computacionais do CRP-MPSP: Cplex x Gurobi

I	Cplex			Gurobi		
	F.O.	Tempo	Gap	F.O.	Tempo	Gap
1A	0	10,94	0,00%	0	6,28	0,00%
2A	1	36,42	0,00%	1	22,95	0,00%
3A	2	19.429,92	0,00%	2	3.260,94	0,00%
4A	9	432.001,34	90,95%	3	432.000,62	66,67%
5A	0	6.079,77	0,00%	0	3.892,21	0,00%
6A	0	21.351,66	0,00%	0	5.883,63	0,00%
7A	-	432.014,87	-	-	432.001,47	-
8A	-	432.005,72	-	-	432.000,85	-
9A	*	*	*	*	*	*
10A	*	*	*	*	*	*
11A	*	*	*	*	*	*
12A	*	*	*	*	*	*

Tabela 7.4 – Resultados Computacionais do CRP-MPSP: Cplex x Gurobi

I	Cplex			Gurobi		
	F.O.	Tempo	Gap	F.O.	Tempo	Gap
1B	0	6,11	0,00%	0	3,30	0,00%
2B	0	12,69	0,00%	0	7,25	0,00%
3B	0	630,81	0,00%	0	273,07	0,00%
4B	0	43.951,89	0,00%	0	3.872,97	0,00%
5B	0	262,83	0,00%	0	552,66	0,00%
6B	0	2.401,53	0,00%	0	1.567,67	0,00%
7B	1146	432.002,89	99,31%	11	61.126,12	0,00%
8B	-	432.002,89	-	-	432.000,88	-
9B	*	*	*	*	*	*
10B	*	*	*	*	*	*
11B	*	*	*	*	*	*
12B	*	*	*	*	*	*

No primeiro, os solvers terminam de construir as instâncias, mas não iniciam o processo de otimização. Isto ocorre, possivelmente por conta da característica combinatória do problema, cuja lista de nós ativos na árvore do Branch & Bound pode consumir grandes quantidades de memória. No segundo tipo de erro de memória, o modelo não termina de ser construído. Isto acontece por falta de memória física. Estes problemas de memória levam os solvers a encerrarem a otimização com uma mensagem de erro. Nestes casos, ajustes nos parâmetros de ênfase de memória não foram exitosos.

Em relação aos resultados obtidos, note que o Gurobi e o Cplex foram capazes de resolver a mesma quantidade de instâncias; entretanto, o Gurobi teve um desempenho melhor em todas as instâncias nas quais uma solução foi encontrada.

Em primeiro lugar, em todas as instâncias nas quais a solução ótima foi encontrada (*Gap* de 0,00%), o Gurobi finalizou a otimização em menos tempo que o Cplex.

Na instância 4A, nenhum dos solvers encontrou a solução ótima dentro do tempo permitido, mas a solução factível encontrada pelo Gurobi foi 24,28% melhor (diferença entre os gaps das soluções encontradas por cada solver).

Na instância 7B, o Gurobi encontrou a solução ótima em 16,98 horas; enquanto isso o Cplex finalizou a otimização porque atingiu o limite de tempo de 5 dias, com uma solução com *gap* de 99,31%. Nas Figuras 7.1 e 7.2 é possível observar o progresso do Cplex e do Gurobi, respectivamente, na instância 7B.

Observe na Figura 7.1, que na instância 7B o Cplex demorou 54.026,77 segundos (15 horas) para encontrar a primeira solução, com valor de 3176. Essa solução é melhorada apenas depois de 57 horas, para 1580; e a otimização termina com a solução de 1146, que foi encontrada após 3 dias de otimização.

```

Nodes                Cuts/
Node Left  Objective  IInf Best Integer  Best Bound  ItCnt  Gap

0 7.2727 18952 7.2727 1

Elapsed time = 22.94 sec. (10011.67 ticks) for 177 of 200 candidates for Gomory cuts

* 0+ 0 3176.0000 7.2727 99.77%

Found incumbent of value 3176.000000 after 54026.77 sec. (16547722.16 ticks)

0 0 7.4000 5615 3176.0000 Cuts: 18 1304877 99.77%

0 0 7.4000 6225 3176.0000 Cuts: 668 3584119 99.77%

0 0 7.9256 5853 3176.0000 Cuts: 500 4415059 99.75%

* 0+ 0 1580.0000 7.9256 99.50%

Found incumbent of value 1580.000000 after 261846.97 sec. (79910156.30 ticks)

* 0+ 0 1389.0000 7.9256 99.43%

Found incumbent of value 1389.000000 after 261941.13 sec. (79948201.93 ticks)

* 0+ 0 1193.0000 7.9256 99.34%

Found incumbent of value 1193.000000 after 262000.94 sec. (79971507.45 ticks)

* 0+ 0 1156.0000 7.9256 99.31%

Found incumbent of value 1156.000000 after 262095.91 sec. (80001997.11 ticks)

* 0+ 0 1146.0000 7.9256 99.31%

Found incumbent of value 1146.000000 after 262100.52 sec. (80005053.05 ticks)

```

Figura 7.1 – Parte do relatório de progresso do Cplex para a instância 7B.

Nesse meio-tempo, observe na Figura 7.2, que na instância 7B o Gurobi demorou 36170 segundos (10 horas) para encontrar a primeira solução inteira, com valor de

22. Essa solução cai pela metade passados um pouco menos de 4 horas, e a otimalidade é comprovada com 16,98 horas de otimização.

Nodes		Current Node			Objective Bounds		Gap	Work	
Expl	Unexpl	Obj	Depth	IntInf	Incumbent	BestBd		It/Node	Time
	0	0	7.27273	0 3693	-	7.27273	-	-	12096s
	0	0	7.28571	0 5984	-	7.28571	-	-	13264s
	0	0	9.00000	0 5011	-	9.00000	-	-	14210s
	0	0	9.04861	0 4994	-	9.04861	-	-	15012s
	0	0	9.15054	0 5193	-	9.15054	-	-	15684s
	0	0	9.17647	0 4810	-	9.17647	-	-	16205s
	0	0	9.66667	0 5635	-	9.66667	-	-	20766s
	0	0	9.66667	0 5502	-	9.66667	-	-	22671s
	0	0	9.66667	0 4483	-	9.66667	-	-	24206s
	0	0	9.66667	0 5007	-	9.66667	-	-	25550s
	0	0	9.81250	0 1986	-	9.81250	-	-	30432s
	0	0	9.82143	0 1943	-	9.82143	-	-	30576s
	0	0	9.82143	0 2183	-	9.82143	-	-	30773s
	0	0	10.00000	0 2209	-	10.00000	-	-	31458s
H	0	0			22.000000	10.00000	54.5%	-	36170s
H	0	0			11.000000	10.00000	9.09%	-	50304s
	0	0	10.00000	0 2143	11.00000	10.00000	9.09%	-	50699s
	0	0	10.00000	0 1314	11.00000	10.00000	9.09%	-	51364s
	0	0	10.00000	0 3124	11.00000	10.00000	9.09%	-	61087s
	0	0	infeasible	0	11.00000	11.00000	0.00%	-	61125s

Figura 7.2 – Parte do relatório de progresso do Gurobi para a instância 7B.

Note que na instância 7B, tanto o Cplex quanto o Gurobi, levaram um tempo considerável para encontrar uma primeira solução factível. Este comportamento também pôde ser observado em algumas das outras instâncias, como na instância 4A, quando resolvida pelo Gurobi (veja a Figura 7.3). Outro ponto observado em relação à convergência dos solvers, é que em algumas instâncias, a primeira solução inteira encontrada é bastante distante da solução ótima, como nas instâncias 7B e 4A, quando resolvidas pelo Cplex (veja as Figuras 7.1 e 7.4).

Nodes		Current Node			Objective Bounds		Gap	Work	
Expl	Unexpl	Obj	Depth	IntInf	Incumbent	BestBd		It/Node	Time
	3009	2223	24.10000	419 1227	-	0.08163	-	7571	42672s
	3380	2535	infeasible	472	-	0.08163	-	6923	42947s
	3836	2887	32.15000	529 415	-	0.12000	-	6291	43196s
*	3894	2863		572	34.000000	0.12000	100%	6201	43196s
	4396	3146	0.14286	10 4247	34.00000	0.12000	100%	5664	44676s
H	4405	2992			31.000000	0.12000	100%	5659	44676s
H	4459	2936			28.000000	0.12000	100%	5620	44676s
	4697	3012	9.84074	165 2322	28.00000	0.12000	100%	5379	45641s
H	4698	2862			25.000000	0.12000	100%	5378	53362s
	4700	2863	25.00000	421 2977	25.00000	0.12000	100%	5375	54570s
H	4700	2720			24.000000	0.12000	100%	5375	55201s
H	4701	2585			23.000000	0.12000	99.5%	5374	58007s
H	4701	2455			21.000000	0.12000	99.4%	5374	59588s

Figura 7.3 – Parte do relatório de progresso do Gurobi para a instância 4A.

Ao comparar os resultados apresentados na Tabela 7.3 com os resultados obtidos nas instâncias em que há mais espaço de manuseio no navio (Tabela 7.4), é possível perceber que mesmo o problema sendo maior — em termos de quantidade de variáveis e restrições — os solvers tiveram um desempenho melhor: encontraram soluções em mais instâncias e utilizaram menos tempo. Isso ocorre porque o modelo tem menos dificuldades para encontrar soluções quando há mais espaço no navio.

Nodes		Objective	IInf	Best Integer	Cuts/		ItCnt	Gap
Node	Left				Best	Bound		
	0	0	0.0000	10577		0.0000	1	
*	0+	0		1103.0000		0.0000		100.00%
Found incumbent of value 1103.000000 after 2391.66 sec. (944322.89 ticks)								
	0	0	0.0000	5937	1103.0000	Cuts: 173	131616	100.00%
	0	0	-0.0000	4561	1103.0000	Cuts: 54	225781	100.00%
	0	0	0.0000	4356	1103.0000	Cuts: 26	329232	100.00%
*	0+	0		918.0000		0.0000		100.00%
Found incumbent of value 918.000000 after 4615.64 sec. (1985918.71 ticks)								
*	0+	0		815.0000		0.0000		100.00%

Figura 7.4 – Parte do relatório de progresso do Cplex para a instância 4A.

De forma geral, observa-se que o tamanho do CRP-MPSP cresce rapidamente e que uma instância com menos de 200 contêineres, ainda que bastante distante de um problema real, já se torna proibitiva para algoritmos exatos.

Observe nas Figuras 7.5 e 7.6 que o crescimento do tempo de resolução de cada instância não acompanha o crescimento do número de variáveis e restrições. Além disso, as instâncias 1B a 8B possuem mais variáveis e restrições do que as instâncias 1A a 8A, e foram resolvidas em menos tempo.

Isso evidencia que o tempo de resolução necessário é específico à características do problema. Com base nisso, pode-se dizer que os itens que mais influenciam na dificuldade de resolução do problema são:

- Quantidade de portos na rota do navio: as instâncias com mais portos na rota do navio demoram mais para serem resolvidas do que as instâncias com menos portos.
- Espaço disponível para manuseio dos contêineres: as instâncias que possuíam mais espaço disponível no navio foram resolvidas em menos tempo do que as instâncias nas quais o espaço no navio era menor.
- Tamanho dos pátios: as instâncias com pátios maiores são mais difíceis de serem resolvidas. Note que as instâncias 9A e 10A possuíam apenas 4 e 6 portos, respectivamente; e mesmo assim não foram resolvidas. O mesmo aconteceu com as instâncias 9B e 10B.

A Figura 7.7 faz um comparativo do tempo necessário para se resolver as instâncias de teste propostas. Em relação às instâncias de 1A a 8A (Figura 7.7(a)), tanto o Cplex quanto o Gurobi apresentaram comportamentos semelhantes; sendo que o Gurobi teve um desempenho levemente superior. Já em relação às instâncias 1B a 8B (Figura 7.7(b)), o Gurobi apresentou um desempenho consideravelmente melhor do que o Cplex.

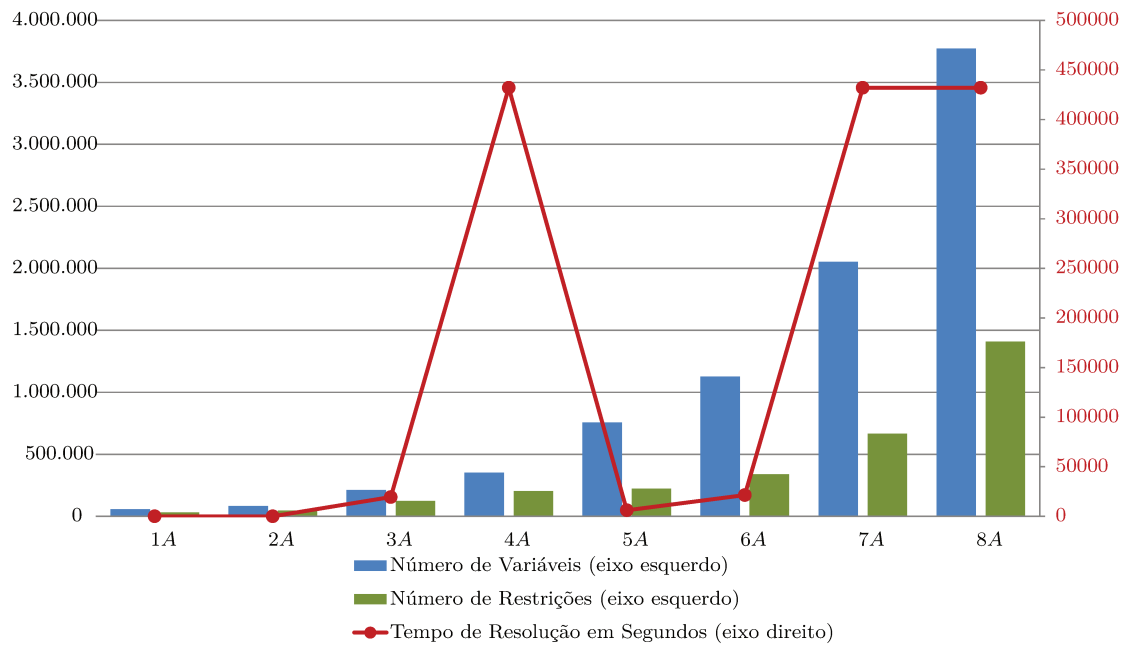


Figura 7.5 – Comparativo entre tempo de resolução e o número de variáveis e restrições das instâncias 1A a 8A obtido pelo Cplex.

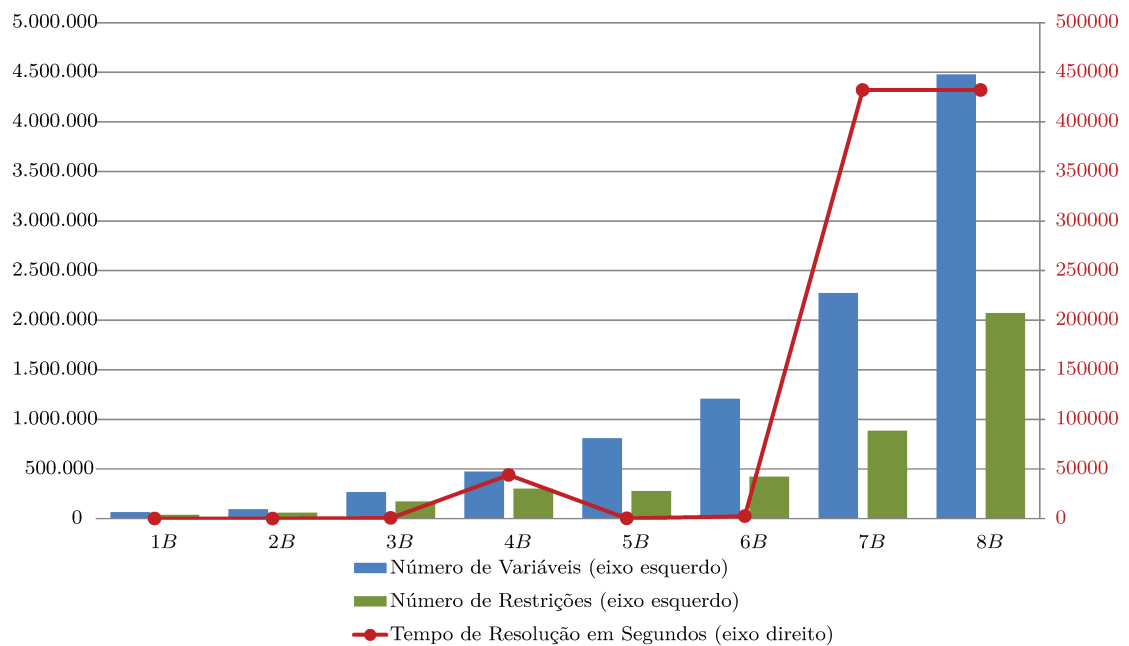
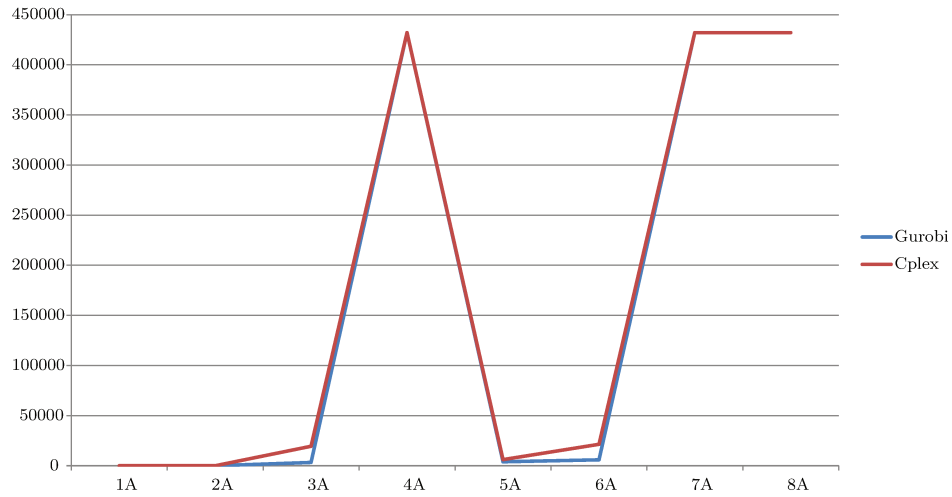
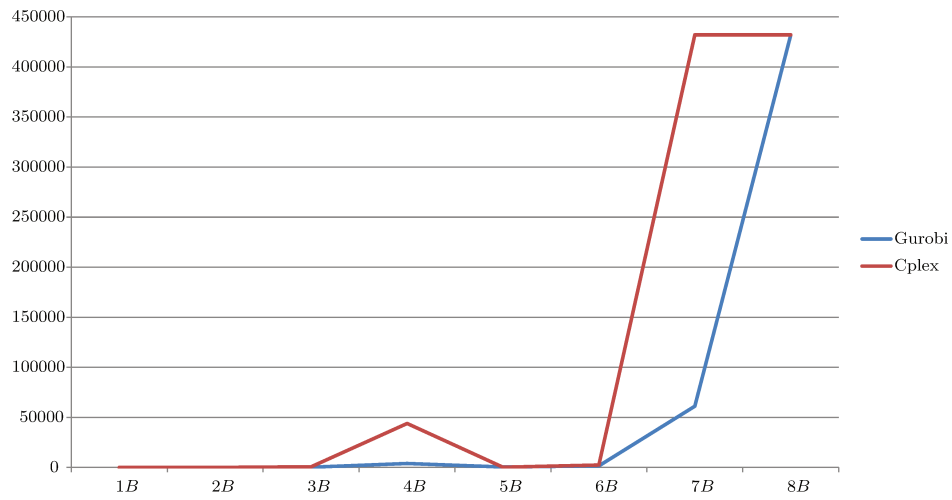


Figura 7.6 – Comparativo entre tempo de resolução e o número de variáveis e restrições das instâncias 1B a 8B obtido pelo Cplex.



(a) Instâncias 1A a 8A



(b) Instâncias 1B a 8B

Figura 7.7 – Cplex x Gurobi: Comparativo dos tempos de resolução.

Na seção seguinte serão apresentados os resultados obtidos com as regras heurísticas HR_1 e HR_2 , que na sequência serão utilizados como *warm-start* pelo modelo do CRP-MPSP.

7.2 Resultados das Regras Heurísticas HR_1 e HR_2

A Tabela 7.5 reporta os resultados obtidos pelas regras heurísticas HR_1 e HR_2 . Os resultados estão apresentados em termos de função objetivo ($F.O.$), que é o número total de remanejamentos, e *Tempo* (em segundos) necessários para resolver as instâncias.

Pela análise da Tabela 7.5, observa-se que ambas as regras heurísticas tomaram um tempo computacional consideravelmente menor do que o método exato; a regra HR_1 foi capaz de resolver quase todas as instâncias de teste em menos de um segundo, enquanto a regra HR_2 foi um pouco mais demorada. No entanto, a regra HR_2 atinigi

Tabela 7.5 – Resultados das regras heurísticas HR_1 e HR_2

I	HR_1		HR_2		I	HR_1		HR_2	
	$F.O.$	$Tempo$	$F.O.$	$Tempo$		$F.O.$	$Tempo$	$F.O.$	$Tempo$
1A	1	0,13	1	53,72	1B	0	0,11	0	21,88
2A	4	0,14	4	111,35	2B	0	0,12	0	18,86
3A	14	0,18	8	282,55	3B	7	0,17	7	432,74
4A	20	0,20	16	804,34	4B	8	0,16	8	946,82
5A	0	0,11	0	38,02	5B	0	0,12	0	51,96
6A	5	0,16	5	230,18	6B	0	0,12	0	42,14
7A	66	0,16	59	840,44	7B	44	0,26	40	1.079,40
8A	63	0,27	48	1.339,20	8B	32	0,35	32	2.257,70
9A	0	0,12	0	108,46	9B	0	0,16	0	195,03
10A	12	0,22	11	882,72	10B	0	0,26	0	227,87
11A	33	0,35	26	2.625,30	11B	0	0,60	0	345,05
12A	82	0,68	66	5.708,80	12B	34	1,48	34	1.489,8

melhores resultados na função objetivo para algumas instâncias, mostrando que postergar o remanejamento pode não ser sempre a melhor escolha.

Na Tabela 7.5, foram resolvidas as mesmas instâncias que o método exato, que são instâncias de pequeno porte. Não obstante, utilizando as regras heurísticas, instâncias maiores também podem ser tratadas.

Em vista disso, para provar o poder das regras heurísticas, novas instâncias de teste são apresentadas na Tabela 7.6. Nestas novas instâncias (I) os pátios possuem 10 linhas (H) por 100 colunas (W) e até 10 mil contêineres. Devido a isso, o navio de cada instância também precisa ter um número maior de linhas (R) e colunas (C). Por fim, P representa a quantidade de portos na rota do navio.

Tabela 7.6 – Instâncias de teste de larga escala.

I	Pátio		Navio		P	Número de Contêineres
	H	W	R	C		
13A	10	100	28	73	4	2.352
14A	10	100	33	91	6	3.801
15A	10	100	39	108	10	6.517
16A	10	100	49	133	15	10.372
13B	10	100	28	146	4	2.352
14B	10	100	33	182	6	3.801
15B	10	100	39	216	10	6.517
16B	10	100	49	266	15	10.372

A Tabela 7.7 apresenta os resultados obtidos com as instâncias de larga escala. Para a regra HR_2 , um tempo computacional máximo de duas horas (7.200 segundos)

ou 10.000 rodadas foram definidas. Quando qualquer uma das condições é atingida, a heurística é interrompida e a melhor solução obtida até então é reportada.

Tabela 7.7 – Resultados computacionais das regras heurísticas em instâncias de larga escala.

I	HR_1		HR_2	
	$O.F.$	$Time$	$O.F.$	$Time$
13A	164	5,56	164	7.200,00
14A	872	16,65	815	7.200,00
15A	2.110	42,86	2.033	7.200,00
16A	4.512	99,25	4.512	7.200,00
13B	45	11,02	45	7.200,00
14B	234	21,31	221	7.200,00
15B	749	52,75	749	7.200,00
16B	1.525	107,18	1.499	7.200,00

Em termos de tempo computacional, a regra heurística HR_1 conseguiu resolver todas as instâncias de larga escala em menos de dois minutos. Enquanto isso, a regra heurística HR_2 atingiu o tempo computacional máximo permitido em todas as instâncias. Apesar disso, ela foi capaz de melhorar a solução de alguns casos. O tempo pode parecer alto, mas os resultados obtidos ainda são bastante superiores aos atingidos pela algoritmo exato em termos de tamanho de instâncias resolvidas e tempo computacional necessário para se obter uma solução.

A seguir são apresentados os resultados computacionais do modelo CRP-MPSP quando a solução obtida pelas regras heurísticas é utilizado como o *warm-start*.

7.3 Resultados do Warm-Start

A construção do *warm-start*, que inclui a tradução da solução da heurística para variáveis binárias, foi efetuada em Python. O modelo com o *warm-start* foi resolvido utilizando o Gurobi 8.0, em um computador com AMD Ryzen 7 1700, 3 GHz, 64 GB RAM, fazendo uso de 16 threads; e utilizando Cplex 12.8, em um cluster IBM de 8 nós, com 20 threads e 120 GB RAM em cada nó. A razão para isso é verificar se o desempenho do Cplex melhora, já que ele apresentou resultados piores do que o Gurobi quando o mesmo computador foi utilizado.

As Tabelas 7.8 e 7.9 mostram os resultados das instâncias descritas nas Tabelas 7.1 e 7.2, respectivamente, quando o *warm-start* é utilizado. A melhor solução obtida pela regra HR_1 ou HR_2 para cada instância é a solução utilizada como solução inicial para o *warm-start*.

Como antes, ao longo dos experimentos foi fixado um tempo computacional máximo de 432.000 segundos (5 dias). Depois disso, o *solver* é interrompido e o resultado alcançado até então é relatado. A primeira coluna, I , mostra o número da instância, a coluna “Solução Inicial” indica o valor da função objetivo da solução obtida pelas regras heurísticas que são usadas como *warm-start*, a coluna “Tempo Heurística” traz a tempo em segundos que a solução inicial foi obtida pelas regras heurísticas HR_1 ou HR_2 . Em seguida, $F.O.$ traz o valor da função objetivo obtida pelo *solver* (Cplex ou Gurobi), a coluna *Tempo* traz o tempo em segundos que o respectivo *solver* levou para resolver a instância, e a coluna *Gap* traz o gap entre a solução encontrada em relação ao best bound.

As colunas preenchidas com um símbolo “*” significam que a memória RAM disponível não foi suficiente para que os solvers processem essas instâncias.

Tabela 7.8 – Resultados computacionais do CRP-MPSP com *warm-start*: Cplex x Gurobi

I	Solução Inicial	Tempo Heurística	Cplex			Gurobi		
			$F.O.$	<i>Tempo</i>	<i>Gap</i>	$F.O.$	<i>Tempo</i>	<i>Gap</i>
1A	1	0,13	0	7,02	0,00%	0	2,02	0,00%
2A	4	0,14	1	50,53	0,00%	1	23,63	0,00%
3A	8	282,55	2	13.052,05	0,00%	2	5.927,86	0,00%
4A	16	804,34	13	432.012,97	100%	8	432.000,72	87,50%
5A	0	0,11	0	15,85	0,00%	0	0,16	0,00%
6A	5	0,16	0	187.348,66	0,00%	0	5.146,15	0,00%
7A	59	840,44	59	432.000,72	89,53%	59	432.001,78	83,05%
8A	48	1.339,2	48	432.000,29	100,00%	48	432.001,02	98,41%
9A	0	0,12	0	1.544,49	0,00%	0	432,34	0,00%
10A	11	882,72	*	*	*	*	*	*
11A	26	2.625,3	*	*	*	*	*	*
12A	68	5.708,8	*	*	*	*	*	*

Tabela 7.9 – Resultados computacionais do CRP-MPSP com *warm-start*: Cplex x Gurobi

I	Solução Inicial	Tempo Heurística	Cplex			Gurobi		
			$F.O.$	<i>Tempo</i>	<i>Gap</i>	$F.O.$	<i>Tempo</i>	<i>Gap</i>
1B	0	0,11	0	0,70	0,00%	0	0,00	0,00%
2B	0	0,12	0	1,08	0,00%	0	0,02	0,00%
3B	7	0,17	0	194,55	0,00%	0	255,71	0,00%
4B	8	0,16	0	4.228,83	0,00%	0	2.548,9	0,00%
5B	0	0,12	0	15,91	0,00%	0	0,17	0,00%
6B	0	0,12	0	25,27	0,00%	0	0,25	0,00%
7B	40	1.079,4	40	432.000,81	78,14%	11	59.820,72	0,00%
8B	32	0,35	32	432.001,41	96,87%	32	432.001,26	96,87%
9B	0	0,16	0	1.445,64	0,00%	0	674,13	0,00%
10B	0	0,26	0	1.661,41	0,00%	0	1.003,85	0,00%
11B	0	0,60	*	*	*	*	*	*
12B	34	1,48	*	*	*	*	*	*

Em geral, usando o *warm-start*, os solvers conseguiram encontrar mais soluções e em menos tempo. Nos casos das instâncias 7A, 8A, 9A, 8B, 9B e 10B, ambos os solvers concluíram a otimização com uma solução. Vale a pena lembrar que, sem uma solução inicial, essas instâncias atingiram o tempo computacional máximo de 432.000 segundos e nenhuma solução factível foi encontrada ou a instância ao menos foi processada.

Pode ser visto que quando a solução inicial também é uma solução ótima e há memória suficiente, o processo de otimização é concluído rapidamente, como foi o caso dos resultados das instâncias 5A, 9A, 1B, 2B, 5B, 6B, 9B, 10B.

Enquanto isso, embora o tempo computacional tenha diminuído na maioria das instâncias, nos casos em que a solução inicial fornecida ainda possui um *gap* muito grande, os solvers podem levar mais tempo para alcançar a solução ótima – como foi o caso das instâncias 2A, 3A no Gurobi e 6A no Cplex; ou sequer melhorar a solução inicial – como foi o caso das instâncias 7B no Cplex, e 7A, 8A e 8B em ambos os solvers. Ainda, no caso da instância 4A, ambos os solvers terminaram a otimização após 5 dias com uma solução pior do que quando o *warm-start* não foi utilizando.

Ao comparar o desempenho entre Cplex e Gurobi nos resultados apresentados nas Tabelas 7.8 e 7.9, o Gurobi concluiu a otimização em tempo menor ou igual do que o Cplex em todas as instâncias (menor em 13 instâncias), exceto nas instâncias 6A e 3B; e chegou a uma solução igual ou melhor do que o Cplex em todas as instâncias (melhor nas instâncias 4A e 7B).

Além disso, nos casos das instâncias 4A, 7A e 8A, apesar de que nenhum dos solvers puderam melhorar a solução inicial fornecida, o Gurobi finalizou a otimização com um *gap* menor do que o Cplex.

Em conclusão, os resultados obtidos mostram que algumas limitações do modelo exato puderam ser superadas com a utilização do *warm-start*, mas ainda apenas instâncias de pequeno a médio porte são solucionáveis de forma exata – o que justifica o uso de métodos heurísticos para resolver instâncias de tamanho real.

Em vista disso, no Capítulo seguinte é proposta uma formulação alternativa que permite melhorar a performance do CRP-MPSP ao reduzir o tamanho do modelo e, consequentemente, resolver instâncias maiores.

8 Uma Formulação Alternativa para o CRP-MPSP

Os resultados obtidos com a utilização do *warm-start* no CRP-MPSP mostraram que foi possível resolver um número maior de instâncias do que quando o *warm-start* não é utilizado. Todavia, este ainda mostrou algumas limitações, como evidenciado no Capítulo 7.

Em vista disso, neste capítulo é feita uma proposta de modificação no modelo do CRP-MPSP, para que este possa resolver instâncias maiores. Este novo modelo é chamado de CRP-MPSP-II.

A modificação proposta consiste basicamente em restringir onde os remanejamentos do problema podem ser efetuados. No caso, o CRP-MPSP-II permite apenas que os remanejamentos necessários sejam efetuados no navio.

Desse modo, o CRP-MPSP-II funcionaria como as regras heurísticas HR_1 e HR_2 , ou seja, dos contêineres que estão no pátio, o modelo deve escolher para retirar um dos que está no topo de cada coluna para carregar no navio. A única opção do CRP-MPSP-II é carregar os contêineres na sua origem e remanejá-los nos portos subsequentes, caso necessário. Esta abordagem pode levar à exclusão de solução ótimas, mas é razoável porque diminui o espaço de busca de soluções.

Com essa modificação, uma vez que remanejamentos no pátio já não são permitidos, a variável x_{ijklnt} se torna desnecessária. Logo, o CRP-MPSP-II pode ser construído da seguinte forma:

$$\text{Minimizar } \sum_{o=1}^{P-1} \sum_{d=o+1}^P \sum_{a=o+1}^{d-1} \sum_{r=1}^R \sum_{c=1}^C w_{oadrc} \quad (8.1)$$

I: *Restrições do pátio*

$$\sum_{i=1}^{W_o} \sum_{j=1}^{H_o} b_{ijnt} + v_{nt} = 1 \quad (8.2)$$

$$n \in \{\Omega_o\}; \quad t \in \{T_o\}; \quad o \in \{1, \dots, P-1\}.$$

$$\sum_{n \in \Omega_o} b_{ijnt} \leq 1 \quad (8.3)$$

$$i \in \{W_o\}; \quad j \in \{H_o\}; \quad t \in \{T_o\}; \quad o \in \{1, \dots, P-1\}$$

$$\sum_{n \in \Omega_o} b_{ijnt} \geq \sum_{n \in \Omega_o} b_{i(j+1)nt} \quad (8.4)$$

$$i \in \{W_o\}; t \in \{T_o\}; j \in \{1, \dots, H_o - 1\}; o \in \{1, \dots, P - 1\}$$

$$b_{ijnt} = b_{ijn(t-1)} - y_{ijn(t-1)} \quad (8.5)$$

$$j \in \{H_o\}; t \in \{2, \dots, T_o\}; o \in \{1, \dots, P - 1\}.$$

$$v_{nt} = \sum_{i=1}^{W_o} \sum_{j=1}^{H_o} \sum_{t'=1}^{t-1} y_{ijnt'} \quad (8.6)$$

$$n \in \{\Omega_o\}; t \in \{2, \dots, T_o\}; o \in \{1, \dots, P - 1\}$$

$$M \left(1 - \sum_{j=1}^{H_o} b_{ijnt} \right) \geq 0 \quad (8.7)$$

$$n \in \Omega_o; i \in \{W_o\}; t \in \{T_o\}; o \in \{1, \dots, P - 1\}.$$

$$1 - \sum_{n \in \Omega_o} b_{i(j+1)nt} \geq \sum_{n \in \Omega_o} y_{ijnt} \quad (8.8)$$

$$j \in \{1, \dots, H_o - 1\}; t \in \{T_o\}; o \in \{1, \dots, P - 1\}$$

II: Restrições de integração

$$\sum_{n \in \Omega_o} v_{nt} = t \quad (8.9)$$

$$t \in \mathbf{T}_o; o \in \{1, \dots, P - 1\}$$

$$v_{n(t+1)} \geq v_{nt} \quad (8.10)$$

$$n \in \Omega_o; t \in \{1, \dots, T_o - 1\}; o \in \{1, \dots, P - 1\}$$

$$\sum_{r=1}^R \sum_{c=1}^C z_{ntrc} = v_{n(t+1)} \quad (8.11)$$

$$n \in \Omega_o; t \in \{1, \dots, T_o - 1\}; o \in \{1, \dots, P - 1\}$$

$$\sum_{o'=1}^{o-1} \sum_{d=o+1}^P \sum_{a=o+1}^d w_{o'darc} + \sum_{d=o+1}^P q_{odrc} + \sum_{n \in \Omega_o} z_{ntrc} \leq 1 \quad (8.12)$$

$$r \in \mathbf{R}; c \in \mathbf{C}; t \in \mathbf{T}_o; o \in \{1, \dots, P - 1\}$$

$$z_{n(t+1)rc} \geq z_{ntrc} \quad (8.13)$$

$$n \in \Omega_o; r \in \mathbf{R}; c \in \mathbf{C}; t \in \{1, \dots, T_o - 1\}; o \in \{1, \dots, P - 1\}$$

$$\sum_{a=o+1}^d w_{odarc} = \sum_{n \in \phi_{o,d}} z_{nT_orc} + q_{odrc} \quad (8.14)$$

$$r \in \mathbf{R}; c \in \mathbf{C}; d \in \{o+1, \dots, P\}; o \in \{1, \dots, P - 1\}$$

$$\sum_{r=1}^R \sum_{c=1}^C z_{nT_orc} = 1 \quad (8.15)$$

$$n \in \Omega_o; o \in \{1, \dots, P - 1\}$$

$$\sum_{n \in \Omega_o} z_{ntrc} + \sum_{o'=1}^{o-1} \sum_{d=o+1}^P \sum_{a=o+1}^d w_{o'darc} + \sum_{d=o+1}^P q_{odrc} \geq \sum_{n \in \Omega_o} z_{nt(r+1)c} \quad (8.16)$$

$$t \in \mathbf{T}_o; r \in \{1, \dots, R - 1\}; c \in \mathbf{C}; o \in \{1, \dots, P - 1\}$$

$$\sum_{o'=1}^{o-1} \sum_{c=1}^C \sum_{r=1}^R w_{o'dorc} = \sum_{r=1}^R \sum_{c=1}^C q_{odrc} \quad (8.17)$$

$$d \in \{o+1, \dots, P\}; \quad o \in \{1, \dots, P-1\}$$

III: Restrições do navio

$$\sum_{a=o+1}^d \sum_{r=1}^R \sum_{c=1}^C w_{odarc} - \sum_{m=1}^{o-1} \sum_{r=1}^R \sum_{c=1}^C w_{mdorc} = F_{od} \quad (8.18)$$

$$o \in \{1, \dots, P-1\}; \quad d \in \{o+1, \dots, P\}$$

$$\sum_{m=1}^o \sum_{d=o+1}^P \sum_{a=o+1}^d w_{mdarc} = u_{orc} \quad (8.19)$$

$$o \in \{1, \dots, P-1\}; \quad r \in \mathbf{R}; \quad c \in \mathbf{C}$$

$$u_{orc} - u_{o(r+1)c} \geq 0 \quad (8.20)$$

$$o \in \{1, \dots, P-1\}; \quad r \in \{1, \dots, R-1\}; \quad c \in \mathbf{C}$$

$$\sum_{o=1}^{d-1} \sum_{e=d}^P w_{oedrc} + \sum_{o=1}^{d-1} \sum_{e=d+1}^P \sum_{a=d+1}^e w_{oea(r+1)c} \leq 1 \quad (8.21)$$

$$d \in \{2, \dots, P\}; \quad r \in \{1, \dots, R-1\}; \quad c \in \mathbf{C}$$

$$\sum_{c=1}^C \sum_{r=\lceil \theta_o / C \rceil + 1}^R u_{orc} = 0 \quad (8.22)$$

$$o \in \{1, \dots, P-1\}$$

A função objetivo no CRP-MPSP-II é minimizar o número total de remaneja-mentos (que acontecem apenas no navio). As restrições (8.2) até (8.22) são as mesmas do modelo CRP-MPSP, mas sem a variável x_{ijklnt} .

Dado isso, a seção seguinte apresenta os resultados computacionais obtidos pelo CRP-MPSP-II quando o mesmo *warm-start* utilizado pelo CRP-MPSP é empregado.

8.1 Resultados Computacionais do CRP-MPSP-II

O CRP-MPSP-II com o *warm-start* foi implementado em Python e executado pelo o Gurobi 8.0, em um computador com AMD Ryzen 7 1700, 3 GHz, 64 GB RAM, fazendo uso de 16 threads; e pelo Cplex 12.8, em um cluster IBM de 8 nós, com 20 threads e 120 GB RAM em cada nó.

Para fins de comparação, as Tabelas 8.1 e 8.2 apresentam o número de restrições geradas pelo CRP-MPSP e pelo CRP-MPSP-II para cada instância I . Note que o CRP-MPSP-II apresenta uma redução significativa nestes termos.

As Tabelas 8.3 e 8.4 mostram os resultados do CRP-MPSP-II ao resolver as instâncias descritas nas Tabelas 7.1 e 7.2, respectivamente, quando o *warm-start* é

Tabela 8.1 – Comparação do tamanho dos modelos.

I	CRP-MPSP		CRP-MPSP-II	
	Quantidade de Variáveis	Quantidade de Restrições	Quantidade de Variáveis	Quantidade de Restrições
1A	57.080	30.373	13.880	13.343
2A	83.924	45.864	22.292	21.378
3A	213.155	123.772	76.355	70.360
4A	351.493	204.102	156.085	127.329
5A	757.955	223.701	107.075	95.529
6A	1.125.877	338.455	164.533	147.883
7A	2.053.070	665.730	364.238	330.370
8A	3.773.715	1.407.996	944.979	849.582
9A	21.861.321	3.340.081	1.179.321	940.433
10A	34,582,630	5.531.912	2.125.030	1.756.152
11A	73,054,437	12.664.333	5.673.237	4.903.713
12A	*	*	10.710.161	9.531.983

Tabela 8.2 – Comparação do tamanho dos modelos.

I	CRP-MPSP		CRP-MPSP-II	
	Quantidade de Variáveis	Quantidade de Restrições	Quantidade de Variáveis	Quantidade de Restrições
1B	63.460	37.095	20.260	19.865
2B	95.516	57.627	33.884	32.925
3B	265.760	172.915	128.960	119.143
4B	473.653	300.916	278.245	223.603
5B	809.660	277.261	158.780	148.189
6B	1.208.629	423.569	247.285	232.133
7B	2.273.640	885.553	584.808	548.793
8B	4.478.055	2.072.990	1.649.319	1.512.182
9B	22.345.497	3.833.009	1.663.497	1.429.413
10B	35.616.724	6.578.547	3.159.124	2.802.787
11B	76.462.917	16.099.967	9.081.717	8.339.347
12B	*	*	18.110.609	16.915.408

utilizado. Estes novos testes empregam o mesmo *warm-start* dos testes apresentados no Capítulo 7.

Novamente, ao longo dos experimentos foi fixado um tempo computacional máximo de 432.000 segundos (5 dias). Depois disso, o *solver* é interrompido e o resultado alcançado até então é relatado. A primeira coluna, I , mostra o número da instância, a coluna “Solução Inicial” indica o valor da função objetivo da solução obtida pelas regras heurísticas que são usadas como *warm-start*, a coluna “Tempo Heurística” traz a tempo em segundos que a solução inicial foi obtida pelas regras heurísticas HR_1 ou HR_2 . Em seguida, $F.O.$ traz o valor da função objetivo obtida pelo *solver* (Cplex ou Gurobi), a coluna *Tempo* traz o tempo em segundos que o respectivo *solver* levou para resolver a instância, e a coluna *Gap* traz o *gap* entre a solução encontrada em relação ao *best bound*.

As células preenchidas com um símbolo “*” significam que a memória RAM disponível não foi suficiente para que os solvers processem essas instâncias.

Tabela 8.3 – Resultados computacionais do CRP-MPSP-II com *warm-start*: Cplex x Gurobi.

I	Solução Inicial	Tempo Heurística	Cplex			Gurobi		
			F.O.	Tempo	Gap	F.O.	Tempo	Gap
1A	1	0,13	0	2,79	0,00%	0	1,3	0,00%
2A	4	0,14	1	35,53	0,00%	1	21,81	0,00%
3A	8	282,55	2	2.480,9	0,00%	2	13.619,76	0,00%
4A	16	804,34	10	432.011,65	100,00%	5	432.000,32	100,00%
5A	0	0,11	0	2,2	0,00%	0	0,02	0,00%
6A	5	0,16	0	2,48,49	0,00%	0	2.986,97	0,00%
7A	59	840,44	16	432.000,66	42,92%	50	432.000,37	82,00%
8A	48	1.339,2	48	432.075,88	100,00%	48	432.000,36	98,40%
9A	0	0,12	0	24,62	0,00%	0	0,34	0,00%
10A	11	882,72	11	432.001,36	100,00%	11	432.000,55	100,00%
11A	26	2.625,3	26	432.016,24	100,00%	26	432.001,31	100,00%
12A	66	5.708,8	66	432.015,35	98,48%	*	*	*

Tabela 8.4 – Resultados computacionais do CRP-MPSP-II com *warm-start*: Cplex x Gurobi.

I	Solução Inicial	Tempo Heurística	Cplex			Gurobi		
			F.O.	Tempo	Gap	F.O.	Tempo	Gap
1B	0	0,11	0	0,26	0,00%	0	0,00	0,00%
2B	0	0,12	0	0,50	0,00%	0	0,00	0,00%
3B	7	0,17	0	210,04	0,00%	0	269,39	0,00%
4B	8	0,16	0	7.883,08	0,00%	0	8.522,76	0,00%
5B	0	0,12	0	2,82	0,00%	0	0,03	0,00%
6B	0	0,12	0	4,68	0,00%	0	0,05	0,00%
7B	40	1.079,4	11	432.000,21	9,09%	11	144.945,64	0,00%
8B	32	0,35	32	432.000,68	96,87%	31	432.000,54	96,77%
9B	0	0,16	0	19,71	0,00%	0	0,39	0,00%
10B	0	0,26	0	47,13	0,00%	0	16,66	0,00%
11B	0	0,60	0	218,53	0,00%	0	137,22	0,00%
12B	34	1,48	34	432.019,72	100,00%	*	*	*

Quando comparados com os resultados do CRP-MPSP, os resultados do CRP-MPSP-II apresentados nas Tabelas 8.3 e 8.4 mostram que os solvers foram capazes de encontrar soluções para um número maior de instâncias. De modo geral, as soluções encontradas foram melhores e menos tempo computacional foi necessário.

Soluções melhores foram encontradas nas instâncias: 4A, 7A e 7B (Cplex e Gurobi) e 8B (Gurobi); e menos tempo computacional foi necessário nas instâncias: 1A, 2A, 5A, 6A, 9A, 2B, 5B, 6B, 9B e 10B (Cplex e Gurobi) e 3A, 1B e 3B (Cplex).

As instâncias 10A, 11A, 12A, 11B e 12B que não tinham sido processadas por falta de memória quando resolvidas pelo CRP-MPSP, puderam ser processadas pelo CRP-MPSP-II. Entretanto, nestas instâncias, nenhum dos solvers foi capaz de melhorar a solução inicial obtida pela abordagem heurística e inserida no modelo como *warm-start* (quando a solução inserida não era a solução ótima).

Comparando o desempenho do Cplex com o Gurobi neste novo modelo, pode-se observar que o Cplex resolveu duas instâncias a mais (12A e 12B) por questões de memória disponível. Em relação ao valor da função objetivo, o Cplex obteve uma solução melhor, na instância 7A, e o Gurobi obteve valores melhores para duas instâncias 4A e 8B. Na instância 8A, ambos obtiveram o mesmo valor para a função objetivo, mas o Gurobi encontrou um *best bound* melhor, e por este motivo, o *gap* da solução final do Gurobi foi menor do que o *gap* da solução do Cplex. Em relação ao tempo computacional, ambos os solvers tiveram desempenhos similares.

Em conclusão, os resultados obtidos mostram que apesar do CRP-MPSP-II, quando comparado com o CRP-MPSP, ter permitido a resolução de mais instâncias de teste em menos tempo computacional, as abordagens exatas ainda ficam limitadas à instâncias de pequeno porte, visto que em alguns casos, os solvers não foram capazes de melhorar a solução inicial, mesmo após 5 dias de otimização.

O Capítulo seguinte apresentará as conclusões desta tese, e possíveis direcionamentos para trabalhos futuros.

9 Conclusão

Nesta tese foi desenvolvido e validado um modelo de otimização que integra os problemas de realocação de contêineres em pátios portuários com o problema do plano de estiva para múltiplos portos, chamado de CRP-MPSP.

O CRP-MPSP é o primeiro modelo exato a representar estas atividades de forma integrada, diferenciando-se assim de uma prática comum e aceita na literatura de tratar cada problema de forma particionada ou simplificada. Por conta disto, foi apresentada uma demonstração através de exemplos de que a resolução do CRP e do MPSP de forma integrada, como proposto, é melhor do que a resolução particionada ou sequencial.

Para resolver o modelo do CRP-MPSP, uma abordagem exata baseada na inserção de *warm-starts* foi proposta. Esta abordagem mostrou-se importante para avançar no processo de otimização das instâncias testadas. Entretanto, apenas instâncias de pequeno e médio porte puderam ser resolvidas. Apesar disso, uma lacuna na literatura foi preenchida, uma vez que decisões de diferentes agentes puderam ser incorporadas. Com isso, foi fornecida uma compreensão valiosa que pode ser usada em diferentes configurações do problema.

Adicionalmente, uma formulação alternativa do CRP-MPSP foi apresentada, chamada de CRP-MPSP-II. Com esta nova formulação algumas limitações do CRP-MPSP foram superadas, mas ainda instâncias pequenas e médias foram resolvidas na otimalidade.

Vale ressaltar ainda que o modelo exato também é importante porque serve de parâmetro de comparação para abordagens heurísticas e meta-heurísticas, como as regras HR_1 e HR_2 que foram desenvolvidas, e que obtiveram soluções ótimas para algumas instâncias de pequeno porte, e boas soluções para instâncias de médio e grande porte dentro de um curto tempo computacional.

Outras importantes contribuições deste trabalho para a literatura são: a construção de instâncias de teste para o modelo exato, que podem ser utilizadas para testar e comparar outras abordagens de solução do CRP-MPSP, a metodologia que transforma a solução no formato matricial obtida pelas regras heurísticas em binário, e também o comparativo dos resultados de diferentes solvers comerciais, que mostram até que tamanho de instâncias é possível resolver com as ferramentas disponíveis atualmente.

Uma possível extensão deste trabalho seria considerar o modelo do CRP-MPSP tri-dimensional, ou ainda aumentar o números de navios na modelagem. Outra linha possível e interessante seria incorporar as incertezas referentes às chegadas e saídas dos

contêineres nos portos.

Relacionado à abordagem proposta, seria interessante buscar outros métodos que gerem soluções iniciais melhores para o problema. Outras formas de resolver o CRP-MPSP também podem ser investigadas, como por exemplo heurísticas baseadas em algoritmos evolutivos.

Referências

- AMBROSINO, D.; ANGHINOLFI, D.; PAOLUCCI, M.; SCIOMACHEN, A. An experimental comparison of different heuristics for the master bay plan problem. *Lecture Notes in Computer Science*, v. 6049, p. 314–325, 2010. Citado na página 28.
- AMBROSINO, D.; SCIOMACHEN, A.; TANFANI, E. Stowing a containership: The master bay plan problem. *Transportation Research Part A: Policy and Practice*, v. 38, n. 2, p. 81–99, 2004. ISSN 09658564. Citado na página 29.
- AMBROSINO, D.; SCIOMACHEN, A.; TANFANI, E. A decomposition heuristics for the container ship stowage problem. *Journal of Heuristics*, v. 12, n. 3, p. 211–233, 2006. Citado na página 28.
- AVRIEL, M.; PENN, M.; SHPIRER, N. Container ship stowage problem: complexity and connection to the coloring of circle graphs. *Discrete Applied Mathematics*, v. 103, n. 1–3, p. 271 – 279, 2000. ISSN 0166-218X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0166218X99002450>>. Citado na página 28.
- AVRIEL, M.; PENN, M.; SHPIRER, N.; WITTEBOON, S. Stowage planning for container ships to reduce the number of shifts. *Annals of Operations Research*, v. 76, n. 0, p. 55–71, 1998. ISSN 0254-5330. Citado 3 vezes nas páginas 28, 40 e 45.
- AYDIN, C. Improved rehandling strategies for container retrieval process. *Master's Thesis, Sabanci University, Turkey*, 2006. Citado na página 23.
- AZEVEDO, A. T.; RIBEIRO, C. M.; NETO, L. L. S. N.; SILVA, M. P. E.; SILVESTRE, M. C. Resolução do problema de carregamento e descarregamento 3D de contêineres em terminais portuários, via representação por regras e algoritmo genético. *GEPROS. Gestão da Produção, Operações e Sistemas*, v. 4, p. 91–110, 2011. Citado na página 13.
- BACCI, T.; MATTIA, S.; VENTURA, P. The bounded beam search algorithm for the block relocation problem. *Computers Operations Research*, v. 103, p. 252 – 264, 2019. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054818302934>>. Citado na página 20.
- BERNHOFEN, D. M.; EL-SAHLI, Z.; KNELLER, R. Estimating the Effects of the Container Revolution on World Trade. *CESifo Working Paper. CESifo, Center for Economic Studies and Ifo Institute*, p. 1–37, 2013. Citado na página 13.
- BIERWIRTH, C.; MEISEL, F. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, v. 202, n. 3, p. 615 – 627, 2010. Citado 3 vezes nas páginas 8, 27 e 33.
- BOTTER, R.; BRINATI, M. Stowage container planning: a model for getting optimal solution. In: *International Conference of Computer Applications in the Automation of Ship-yard Operation and Ship Design*. [S.l.: s.n.], 1991. p. 217–229. Citado na página 28.

BRAEKERS, K.; CARIS, A.; JANSSENS, G. Integrated planning of loaded and empty container movements. *OR Spectrum*, v. 35, n. 2, p. 457–478, 2013. ISSN 0171-6468. Citado na página 33.

BROOKS, M. R.; SCHELLINCK, T. Measuring port effectiveness in user service delivery: What really determines users' evaluations of port service delivery? *Research in Transportation Business Management*, v. 8, p. 87 – 96, 2013. ISSN 2210-5395. Port Performance and Strategy. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2210539513000321>>. Citado na página 32.

CARLO, H. J.; VIS, I. F.; ROODBERGEN, K. J. Storage yard operations in container terminals: Literature overview, trends, and research directions. *European Journal of Operational Research*, v. 235, n. 2, p. 412 – 430, 2014. ISSN 0377-2217. Maritime Logistics. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221713008771>>. Citado 3 vezes nas páginas 8, 16 e 26.

CASERTA, M.; SCHWARZE, S.; VOSS, S. A new binary description of the blocks relocation problem and benefits in a look ahead heuristic. In: COTTA, C.; COWLING, P. (Ed.). *Evolutionary Computation in Combinatorial Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 37–48. ISBN 978-3-642-01009-5. Citado na página 24.

CASERTA, M.; SCHWARZE, S.; VOSS, S. A mathematical formulation and complexity considerations for the blocks relocation problem. *European Journal of Operational Research*, Elsevier B.V., v. 219, n. 1, p. 96–104, 2012. ISSN 03772217. Disponível em: <<http://dx.doi.org/10.1016/j.ejor.2011.12.039>>. Citado 7 vezes nas páginas 19, 20, 22, 23, 24, 40 e 59.

CASERTA, M.; VOSS, S. Corridor selection and fine tuning for the corridor method. In: STÜTZLE, T. (Ed.). *Learning and Intelligent Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 163–175. ISBN 978-3-642-11169-3. Citado na página 20.

CASERTA, M.; VOSS, S.; SNIEDOVICH, M. Applying the corridor method to a blocks relocation problem. *OR Spectrum*, v. 33, n. 4, p. 915–929, Oct 2011. ISSN 1436-6304. Disponível em: <<https://doi.org/10.1007/s00291-009-0176-5>>. Citado 2 vezes nas páginas 20 e 24.

CHANG, D.; JIANG, Z.; YAN, J. H. W. Integrating berth allocation and quay crane assignments. *Transportation Research Part E*, v. 46, n. 6, p. 975–990, 2010. Citado na página 33.

CHEN, C.; LEE, S.; SHEN, Q. An analytical model for the container loading problem. *European Journal of Operational Research*, v. 80, n. 1, p. 68 – 76, 1995. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/037722179400002T>>. Citado na página 28.

CHEN, L.; LU, Z. The storage location assignment problem for outbound containers in a maritime terminal. *International Journal of Production Economics*, v. 135, n. 1, p. 73 – 80, 2012. ISSN 0925-5273. Advances in Optimization and Design of Supply Chains. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925527310003579>>. Citado na página 17.

- DING, D.; CHOU, M. C. Stowage planning for container ships: A heuristic algorithm to reduce the number of shifts. *European Journal of Operational Research*, v. 246, n. 1, p. 242 – 249, 2015. Citado na página 42.
- DUBROVSKY, O.; LEVITIN, G.; PENN, M. A genetic algorithm with a compact solution encoding for the container ship stowage problem. *Journal of Heuristics*, v. 8, n. 6, p. 585–599, 2002. ISSN 13811231. Citado na página 28.
- EXPÓSITO-IZQUIERDO, C.; MELIÁN-BATISTA, B.; MORENO-VEGA, J. M. An exact approach for the blocks relocation problem. *Expert Systems with Applications*, v. 42, n. 17, p. 6408 – 6422, 2015. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417415002511>>. Citado na página 24.
- FAN, L.; LOW, M.; YING, H.; JING, H.; MIN, Z.; AYE, W. Stowage planning of large containership with tradeoff between crane workload balance and ship stability. In: *Proceedings of the International MultiConference of Engineers and Computers Scientists III*. [S.l.: s.n.], 2010. p. 1–7. Citado na página 28.
- FIRMINO, A. da S.; SILVA, R. M. de A.; TIMES, V. C. A reactive grasp metaheuristic for the container retrieval problem to reduce crane's working time. *Journal of Heuristics*, v. 25, n. 2, p. 141–173, Apr 2019. ISSN 1572-9397. Disponível em: <<https://doi.org/10.1007/s10732-018-9390-0>>. Citado 2 vezes nas páginas 20 e 26.
- FORSTER, F.; BORTFELDT, A. A tree search procedure for the container relocation problem. *Computers Operations Research*, v. 39, n. 2, p. 299 – 309, 2012. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054811001055>>. Citado na página 20.
- GALLE, V.; BARNHART, C.; JAILLET, P. A new binary formulation of the restricted container relocation problem based on a binary encoding of configurations. *European Journal of Operational Research*, v. 267, n. 2, p. 467 – 477, 2018. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221717310640>>. Citado 2 vezes nas páginas 24 e 26.
- HA, M.-H.; YANG, Z.; LAM, J. S. L. Port performance in container transport logistics: A multi-stakeholder perspective. *Transport Policy*, v. 73, p. 25 – 40, 2019. ISSN 0967-070X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0967070X17307412>>. Citado na página 17.
- HA, M.-H.; YANG, Z.; LAM, J. S. L. Port performance in container transport logistics: A multi-stakeholder perspective. *Transport Policy*, v. 73, p. 25 – 40, 2019. ISSN 0967-070X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0967070X17307412>>. Citado na página 32.
- HENDRIKS, M. P. M.; LEFEBER, E.; UDDING, J. T. Simultaneous berth allocation and yard planning at tactical level. *OR Spectrum*, v. 35, n. 2, p. 441–456, 2013. ISSN 0171-6468. Citado na página 33.
- HUSSEIN, M.; PETERING, M. Global retrieval heuristic and genetic algorithm in block relocation problem. *62nd IIE Annual Conference and Expo 2012*, p. 1489–1498, 01 2012. Citado na página 20.

- IMAI, A.; CHEN, H. C.; NISHIMURA, E.; PAPADIMITRIOU, S. The simultaneous berth and quay crane allocation problem. *Transportation Research Part E*, v. 44, n. 5, p. 900–920, 2008. Citado na página 33.
- IMAI, A.; SASAKI, K.; NISHIMURA, E.; PAPADIMITRIOU, S. Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard stacks. *European Journal of Operational Research*, v. 171, n. 2, p. 373–389, 2006. ISSN 03772217. Citado 2 vezes nas páginas 28 e 29.
- IRIS, C.; PACINO, D. A survey on the ship loading problem. In: _____. *Computational Logistics: 6th International Conference, ICCL 2015, Delft, The Netherlands, September 23–25, 2015, Proceedings*. Cham: Springer International Publishing, 2015. p. 238–251. ISBN 978-3-319-24264-4. Disponível em: <http://dx.doi.org/10.1007/978-3-319-24264-4_17>. Citado 2 vezes nas páginas 28 e 33.
- IRIS Çağatay; CHRISTENSEN, J.; PACINO, D.; ROPKE, S. Flexible ship loading problem with transfer vehicle assignment and scheduling. *Transportation Research Part B: Methodological*, v. 111, p. 113 – 134, 2018. ISSN 0191-2615. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0191261517308196>>. Citado 3 vezes nas páginas 30, 31 e 33.
- JI, M.; GUO, W.; ZHU, H.; YANG, Y. Optimization of loading sequence and rehandling strategy for multi-quay crane operations in container terminals. *Transportation Research Part E: Logistics and Transportation Review*, v. 80, p. 1 – 19, 2015. ISSN 1366-5545. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1366554515001040>>. Citado 2 vezes nas páginas 19 e 34.
- JIN, B.; ZHU, W.; LIM, A. Solving the container relocation problem by an improved greedy look-ahead heuristic. *European Journal of Operational Research*, v. 240, n. 3, p. 837 – 847, 2015. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221714006043>>. Citado 2 vezes nas páginas 8 e 19.
- JOVANOVIĆ, R.; TANAKA, S.; NISHI, T.; VOSS, S. A grasp approach for solving the blocks relocation problem with stowage plan. *Flexible Services and Manufacturing Journal*, Jul 2018. ISSN 1936-6590. Disponível em: <<https://doi.org/10.1007/s10696-018-9320-3>>. Citado na página 34.
- JOVANOVIĆ, R.; TUBA, M.; VOß, S. An efficient ant colony optimization algorithm for the blocks relocation problem. *European Journal of Operational Research*, v. 274, n. 1, p. 78 – 90, 2019. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221718308208>>. Citado na página 20.
- JOVANOVIĆ, R.; VOß, S. A chain heuristic for the blocks relocation problem. *Computers Industrial Engineering*, v. 75, p. 79 – 86, 2014. ISSN 0360-8352. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0360835214001909>>. Citado na página 23.
- KANG, J.-G.; KIM, Y.-D. Stowage planning in maritime container transportation. *Journal of the Operational Research Society*, v. 53, n. 4, p. 415–426, Apr 2002. ISSN 1476-9360. Disponível em: <<http://dx.doi.org/10.1057/palgrave.jors.2601322>>. Citado na página 28.

- KIM, K. H.; HONG, G. P. A heuristic rule for relocating blocks. *Computers and Operations Research*, v. 33, n. 4, p. 940–954, 2006. ISSN 03050548. Citado 4 vezes nas páginas 20, 21, 23 e 34.
- KIM, K. H.; LEE, H. Container terminal operation: Current trends and future challenges. In: LEE, C. Y.; MENG, Q. (Ed.). *Handbook of Ocean Container Transport Logistics*. Nova York: Springer, 2015. cap. 2, p. 43–74. Citado na página 26.
- LEE, C. Y.; SONG, D. P. Ocean container transport in global supply chains: Overview and research opportunities. p. 442–474, 05 2017. Citado na página 26.
- LEE, Y.; HSU, N. Y. An optimization model for the container pre-marshalling problem. *Computers and Operations Research*, v. 34, n. 11, p. 3295–3313, 2007. ISSN 03050548. Citado na página 25.
- LEE, Y.; LEE, Y. J. A heuristic for retrieving containers from a yard. *Computers and Operations Research*, Elsevier, v. 37, n. 6, p. 1139–1147, 2010. ISSN 03050548. Disponível em: <<http://dx.doi.org/10.1016/j.cor.2009.10.005>>. Citado 3 vezes nas páginas 21, 23 e 26.
- LEHNFELD, J.; KNUST, S. Loading , unloading and premarshalling of stacks in storage areas : Survey and classification. *European Journal of Operational Research*, Elsevier B.V., v. 239, n. 2, p. 297–312, 2014. ISSN 0377-2217. Disponível em: <<http://dx.doi.org/10.1016/j.ejor.2014.03.011>>. Citado 2 vezes nas páginas 26 e 28.
- MEISEL, F.; BIERWIRTH, C. Heuristics for the integration of crane productivity in the berth allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, v. 45, n. 1, p. 196 – 209, 2009. Citado na página 33.
- MEISEL, F.; WICHMANN, M. Container sequencing for quay cranes with internal reshuffles. *OR Spectrum*, v. 32, n. 3, p. 569–591, Jul 2010. ISSN 1436-6304. Disponível em: <<https://doi.org/10.1007/s00291-009-0191-6>>. Citado na página 31.
- MONACO, M. F.; SAMMARRA, M.; SORRENTINO, G. The terminal-oriented ship stowage planning problem. *European Journal of Operational Research*, v. 239, n. 1, p. 256 – 265, 2014. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221714004536>>. Citado na página 28.
- MONACO, M. F.; SAMMARRA, M.; SORRENTINO, G. The terminal-oriented ship stowage planning problem. *European Journal of Operational Research*, Elsevier B.V., v. 239, n. 1, p. 256–265, 2014. ISSN 03772217. Disponível em: <<http://dx.doi.org/10.1016/j.ejor.2014.05.030>>. Citado 4 vezes nas páginas 25, 30, 32 e 33.
- MURTY, K. G.; LIU, J.; WAN, Y. W.; LINN, R. A decision support system for operations in a container terminal. *Decision Support Systems*, v. 39, n. 3, p. 309–332, 2005. ISSN 01679236. Citado na página 21.
- NISHI, T.; KONISHI, M. An optimisation model and its effective beam search heuristics for floor-storage warehousing systems. *International Journal of Production Research*, Taylor Francis, v. 48, n. 7, p. 1947–1966, 2010. Disponível em: <<https://doi.org/10.1080/00207540802603767>>. Citado na página 20.

- PACINO, D.; DELGADO, A.; JENSEN, R. M.; BEBBINGTON, T. Fast generation of near-optimal plans for eco-efficient stowage of large container vessels. In: _____. *Computational Logistics: Second International Conference, ICCL 2011, Hamburg, Germany, September 19-22, 2011. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 286–301. ISBN 978-3-642-24264-9. Disponível em: <http://dx.doi.org/10.1007/978-3-642-24264-9_22>. Citado na página 28.
- PARREÑO-TORRES, C.; ALVAREZ-VALDES, R.; PARREÑO, F. Solution strategies for a multiport container ship stowage problem. *Mathematical Problems in Engineering*, v. 2019, p. 1–12, 05 2019. Citado 2 vezes nas páginas 30 e 33.
- PETERING, M. E. H.; HUSSEIN, M. I. A new mixed integer program and extended look-ahead heuristic algorithm for the block relocation problem. *European Journal of Operational Research*, Elsevier B.V., v. 231, n. 1, p. 120–130, 2013. ISSN 03772217. Disponível em: <<http://dx.doi.org/10.1016/j.ejor.2013.05.037>>. Citado na página 23.
- PETERING, M. E. H.; SEO, J.; LEE, C. Optimal control of material handling systems: Research project final report. *Department of Industrial and Systems Engineering, National University of Singapore*, 2005. Citado na página 23.
- QUISPE, K. E. Y.; LINTZMAYER, C. N.; XAVIER, E. C. An exact algorithm for the blocks relocation problem with new lower bounds. *Computers Operations Research*, v. 99, p. 206 – 217, 2018. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054818301710>>. Citado na página 24.
- RAHMAN, N. S. F. A.; ISMAIL, A.; LUN, V. Y. Preliminary study on new container stacking/storage system due to space limitations in container yard. *Maritime Business Review*, v. 1, n. 1, p. 21–39, 2016. Disponível em: <<https://doi.org/10.1108/MABR-03-2016-0004>>. Citado na página 33.
- SCIOMACHEN, A.; TANFANI, E. A 3D-BPP approach for optimising stowage plans and terminal productivity. *European Journal of Operational Research*, v. 183, n. 3, p. 1433–1446, 2007. ISSN 03772217. Citado na página 28.
- SILVA, M. de Melo da; ERDOGAN, G.; BATTARRA, M.; STRUSEVICH, V. The block retrieval problem. *European Journal of Operational Research*, v. 265, n. 3, p. 931 – 950, 2018. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221717307798>>. Citado na página 25.
- SILVA, M. de Melo da; TOULOUSE, S.; CALVO, R. W. A new effective unified model for solving the pre-marshalling and block relocation problems. *European Journal of Operational Research*, v. 271, n. 1, p. 40 – 56, 2018. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221718303795>>. Citado na página 25.
- STAHLBOCK, R.; VOSS, S. Operations research at container terminals: A literature update. v. 30, p. 1–52, 04 2008. Citado na página 26.
- STEENKEN, D.; VOSS, S.; STAHLBOCK, R. Container terminal operation and operations research - A classification and literature review. *OR Spectrum*, v. 26, p. 3–49, 2004. ISSN 01716468. Citado 2 vezes nas páginas 17 e 26.

- TANAKA, S.; TAKII, K. A faster branch-and-bound algorithm for the block relocation problem. *IEEE Transactions on Automation Science and Engineering*, v. 13, n. 1, p. 181–190, Jan 2016. ISSN 1545-5955. Citado 2 vezes nas páginas 24 e 25.
- TANAKA, S.; VOß, S. An exact algorithm for the block relocation problem with a stowage plan. *European Journal of Operational Research*, 2019. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221719304898>>. Citado na página 34.
- TANG, L.; ZHAO, R.; LIU, J. Models and algorithms for shuffling problems in steel plants. *Naval Research Logistics (NRL)*, Wiley Subscription Services, Inc., A Wiley Company, v. 59, n. 7, p. 502–524, 2012. ISSN 1520-6750. Disponível em: <<http://dx.doi.org/10.1002/nav.21503>>. Citado 2 vezes nas páginas 22 e 26.
- TING, C.-J.; WU, K.-C. Optimizing container relocation operations at container yards with beam search. *Transportation Research Part E: Logistics and Transportation Review*, v. 103, p. 17 – 31, 2017. ISSN 1366-5545. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1366554516308134>>. Citado na página 20.
- TRICOIRE, F.; SCAGNETTI, J.; BEHAM, A. New insights on the block relocation problem. *Computers and Operations Research*, v. 89, p. 127 – 139, 2018. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054817302174>>. Citado 4 vezes nas páginas 8, 20, 21 e 24.
- UNCTAD/RMT. *Review of Maritime Transport - 2017*. Gênova: United Nations Publication, 2017. 114 p. ISSN 0566-7682. ISBN 978-92-1-112922-9. Disponível em: <http://unctad.org/en/PublicationsLibrary/rmt2017_en.pdf>. Citado na página 13.
- UNCTADstat. *United Nations Conference on Trade and Development - Container port throughput report*. 2018. Disponível em: <<http://unctadstat.unctad.org/wds/TableView/tableView.aspx?ReportId=13321>>. Acesso em: 13 junho 2018. Citado na página 13.
- UNCTADstat. *United Nations Conference on Trade and Development - Maritime Profile: Brazil*. 2018. Disponível em: <<http://unctadstat.unctad.org/CountryProfile/MaritimeProfile/en-GB/076/index.html>>. Acesso em: 13 junho 2018. Citado na página 13.
- WAN, Y. W.; LIU, J.; TSAI, P. C. The assignment of storage locations to containers for a container stack. *Naval Research Logistics (NRL)*, Wiley Subscription Services, Inc., A Wiley Company, v. 56, n. 8, p. 699–713, 2009. ISSN 1520-6750. Disponível em: <<http://dx.doi.org/10.1002/nav.20373>>. Citado 5 vezes nas páginas 8, 19, 21, 22 e 26.
- WILSON, A. I. D.; ROACH, P. A. Container Stowage Planning : A Methodology for Generating Computerised Solutions. *Journal of the Operational Research Society*, v. 51, n. 11, p. 1248–1255, 2000. Citado 2 vezes nas páginas 28 e 32.
- WILSON, I.; ROACH, P. Principles of combinatorial optimization applied to container-ship stowage planning. *Journal of Heuristics*, v. 5, n. 4, p. 403–418, 1999. Citado na página 28.
- WILSON, I.; ROACH, P.; WARE, J. Container stowage pre-planning: using search to generate solution, a case of study. *Knowledge Based Systems*, v. 14, p. 137–147, 2001. Citado 2 vezes nas páginas 28 e 29.

WU, K.-C.; TING, C.-J. A beam search algorithm for minimizing reshuffle operations at container yards. *Proceedings of the International Conference on Logistics and Maritime Systems*, 01 2010. Citado na página 20.

YANG, C.; WANG, X.; LI, Z. An optimization approach for coupling problem of berth allocation problem and quay crane assignment in container terminal. *Computers & Industrial Engineering*, v. 63, n. 1, p. 243–253, 2012. Citado na página 33.

ZEHEMNER, E.; CASERTA, M.; FEILLET, D.; SCHWARZE, S.; VOSS, S. An improved mathematical formulation for the blocks relocation problem. *European Journal of Operational Research*, Elsevier Ltd., v. 245, n. 2, p. 415–422, 2015. ISSN 03772217. Disponível em: <<http://dx.doi.org/10.1016/j.ejor.2015.03.032>>. Citado 2 vezes nas páginas 23 e 40.

ZHEN, L.; JIANG, X.; LEE, L. H.; CHEW, E. P. A review on yard management in container terminals. *Industrial Engineering and Management Systems*, v. 12, 12 2013. Citado 3 vezes nas páginas 13, 17 e 18.

ZHEN, L.; XU, Z.; WANG, K.; DING, Y. Multi-period yard template planning in container terminals. *Transportation Research Part B: Methodological*, v. 93, p. 700 – 719, 2016. ISSN 0191-2615. Maritime Logistics. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0191261515002672>>. Citado na página 33.

ZHU, W.; QIN, H.; LIM, A.; ZHANG, H. Iterative deepening a* algorithms for the container relocation problem. *IEEE Transactions on Automation Science and Engineering*, v. 9, n. 4, p. 710–722, Oct 2012. ISSN 1545-5955. Citado na página 24.

ÜNLÜYURT, T.; AYDIN, C. Improved rehandling strategies for the container retrieval process. *Journal of Advanced Transportation*, v. 46, n. 4, p. 378–393, 2012. ISSN 2042-3195. Disponível em: <<http://dx.doi.org/10.1002/atr.1193>>. Citado 2 vezes nas páginas 23 e 26.