

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO TECNOLÓGICO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA CIVIL  
MESTRADO EM ENGENHARIA CIVIL

JONAS PALUCI BARBOSA

**DISSERTAÇÃO DE MESTRADO**

**META-HEURÍSTICA APLICADA AO PLANEJAMENTO DA FILA DE NAVIOS  
INTEGRADO À DESIGNAÇÃO DE ÁREAS DE PÁTIO DE GRANEL PARA  
ATENDER AO PLANO DE PRODUÇÃO DE UMA USINA DE PELOTIZAÇÃO**

VITÓRIA  
2020

JONAS PALUCI BARBOSA

**DISSERTAÇÃO DE MESTRADO**

**META-HEURÍSTICA APLICADA AO PLANEJAMENTO DA FILA DE NAVIOS  
INTEGRADO À DESIGNAÇÃO DE ÁREAS DE PÁTIO DE GRANEL PARA  
ATENDER AO PLANO DE PRODUÇÃO DE UMA USINA DE PELOTIZAÇÃO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do título de Mestre em Engenharia Civil.

Orientador: Prof. Dr. Rodrigo de Alvarenga Rosa.

VITÓRIA  
2020

Ficha catalográfica disponibilizada pelo Sistema Integrado de  
Bibliotecas - SIBI/UFES e elaborada pelo autor

---

B238           Barbosa, Jonas Paluci, 1994-  
m            Meta-heurística aplicada ao planejamento da fila de navios  
              integrado à designação de áreas de pátio de granel para atender  
              ao plano de produção de uma usina de pelotização / Jonas Paluci  
              Barbosa. - 2020.  
              105 f. : il.

              Orientador: Rodrigo de Alvarenga Rosa.  
              Dissertação (Mestrado em Engenharia Civil) - Universidade  
              Federal do Espírito Santo, Centro Tecnológico.

              1. Terminais marítimos. 2. Cadeia de logística integrada. 3.  
              Logística Naval. 4. Otimização Combinatória. 5. Heurística. I. Rosa,  
              Rodrigo de Alvarenga. II. Universidade Federal do Espírito  
              Santo. Centro Tecnológico. III. Título.

CDU: 624

---

**META-HEURÍSTICA APLICADA AO PLANEJAMENTO DA FILA DE NAVIOS  
INTEGRADO À DESIGNAÇÃO DE ÁREAS DE PÁTIO DE GRANEL PARA  
ATENDER AO PLANO DE PRODUÇÃO DE UMA USINA DE PELOTIZAÇÃO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito para obtenção do título de Mestre em Engenharia Civil.

Aprovado em 25 de março de 2020 por:

---

Prof. Dr. Rodrigo de Alvarenga Rosa  
Doutor em Engenharia Elétrica  
Orientador - UFES

---

Prof. Dr. Macksuel Soares de Azevedo  
Doutor em Engenharia de Estruturas  
Examinador Interno - UFES

---

Prof. Dr. Rafael Buback Teixeira  
Doutor em Engenharia de Produção  
Examinador Externo - IFES

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus por ter me dado forças para concluir mais uma jornada.

Agradeço a minha família, em especial à minha mãe, pelo gigantesco apoio, me incentivando a seguir mesmo nos momentos mais difíceis.

Agradeço aos colegas de mestrado, tanto pelos momentos de descontração quanto pelo suporte que foi me dado.

Agradeço ao Valdir da Silva Corrêa, o Júnior, que tive o prazer de conhecer e que se tornou um grande amigo. Pelos conselhos, opiniões, conversas descontraídas regadas à café e, principalmente pela paciência em ouvir os meus desabafos, meus sinceros agradecimentos.

Por fim, agradeço ao Prof. Dr. Rodrigo de Alvarenga Rosa, que aceitou me orientar, me guiando para a conclusão desta etapa importante em minha vida, acreditando na minha capacidade e sendo paciente e compreensivo nos momentos de dificuldade. Ter a oportunidade de trabalhar com um professor e pesquisador engajado e dedicado ao que faz, sempre disposto a ajudar os seus alunos, é um prazer e uma lição para quem pensa em seguir a carreira acadêmica.

## RESUMO

A pelota de minério de ferro é um dos principais insumos da indústria siderúrgica. Este commodity é produzido em grandes volumes e armazenado em pátios de estocagem até o momento de seu transporte para o mercado consumidor. Neste trabalho, é considerado o modo de transporte marítimo. Como o processo é contínuo e ininterrupto, antes de iniciar a produção, deve haver espaço em pátios próprios para estocar o material produzido. A paralisação da usina por falta de espaço no pátio acarreta impactos financeiros significativos. Assim, demanda-se uma gestão eficiente do pátio de armazenagem. Para liberar a área do pátio, deve-se planejar a fila de navios no porto a fim de que a retirada de carga do pátio tenha a frequência necessária para não interromper a produção. Portanto, deve-se realizar o planejamento da fila de navios integrado à designação de áreas do pátio de armazenagem de granel para atender ao plano de produção da usina. Na literatura, os problemas que abordam o planejamento da atracação dos navios e a designação das áreas de pátio são denominados, respectivamente Problema de Alocação de Berços (PAB) e *Yard Assignment Problem* (YAP). A integração destes problemas, considerando o atendimento ao plano de produção de uma usina de pelotização, foi encontrada em apenas uma pesquisa que propôs um modelo matemático. No entanto, dada a natureza do problema, classificado como *NP-Hard*, a resolução do modelo por meio de *solver* comercial não conseguiu resolver instâncias de grande porte em um tempo computacional aceitável para uma aplicação prática. Assim, nesta dissertação foi desenvolvida uma meta-heurística para resolver o planejamento da fila de navios integrado à designação de áreas do pátio de armazenagem, visando atender o plano de produção de uma usina de pelotização. A meta-heurística foi capaz de resolver todas as instâncias em menos de três minutos, obtendo resultados próximos do ótimo nas instâncias de pequeno porte e superando a solução gerada pelo *solver* na maioria das instâncias de grande porte. A qualidade das soluções obtidas, aliada aos tempos de execução significativamente baixos, atestam a aplicabilidade da meta-heurística desenvolvida.

Palavras-chave: Problema de Alocação de Berços; Problema de Designação de Áreas do Pátio; Planejamento integrado; Portos de carga a granel; Usinas de pelotização.

## **ABSTRACT**

The iron ore pellet is one of the main inputs of the steel industry. This commodity is produced in large volumes in a continuous and uninterrupted way, being stored in yard locations until its transportation to the consumer market. In this work, the maritime mode of transport is considered. Given that the process is continuous and uninterrupted, before starting production, there should be enough space in the storage yards for the produced material. Stopping the production due to lack of available area in the yard entails significant financial impacts. Thus, efficient management of the storage yard is required. In order to free up space in the yard, allowing an uninterrupted production, planning the port line-up is necessary to withdraw the cargo from the yard at the required frequency. Therefore, the port line-up planning should be integrated to the assignment of storage yard locations for bulk cargo, so that the production plan of the pelletizing plant is met. In the literature, the problems that concerns the planning of port line-up and the assignment of yard locations are denominated, respectively, Berth Allocation Problem (BAP) and Yard Assignment Problem (YAP). The integration of both problems, considering the fulfillment of the production plan in a pelletizing plant, was only found in one research, which proposed a mathematical model. However, given the nature of the problem, classified as NP-Hard, solving the problem by means of a commercial solver was not able to solve large instances in a reasonable time for practical usage. Thus, in this research, a meta-heuristic was developed to solve the port line-up planning integrated to the assignment of storage yard locations, aiming to fulfill the production plan of the pelletizing plant. The proposed meta-heuristic was able to solve all instances in less than three minutes, obtaining, at least, sub-optimal solutions in small instances and surpassing the solver generated solution in most large instances. The quality of the solutions, coupled with significantly low execution times, allows to attest the applicability of the developed meta-heuristic.

**Keywords:** Berth Allocation Problem; Yard Assignment Problem; Integrated planning; Bulk cargo ports; Pelletizing plants.

## LISTA DE FIGURAS

Figura 1 – Representação gráfica do PAB discreto, contínuo e híbrido .....	19
Figura 2 - Principais parâmetros e variáveis de decisão do modelo matemático da parte sobre o PAB .....	41
Figura 3 - Principais parâmetros e variáveis de decisão do modelo matemático da parte sobre o Pátio .....	42
Figura 4 - Etapas do projeto .....	52
Figura 5 - Vista aérea da planta de processamento da Samarco em Anchieta-ES...	56
Figura 6 - Pátios de armazenagem de pelotas da Samarco.....	57
Figura 7 - Representação gráfica das balizas do pátio real da Samarco .....	58
Figura 8 - Representação gráfica das balizas do pátio real da Samarco .....	64
Figura 9 – Estrutura <i>Ship</i> .....	66
Figura 10 – Estrutura <i>Pile</i> .....	67
Figura 11 – Estrutura <i>Solution</i> .....	67
Figura 12 – Estrutura <i>Sol_Pile</i> .....	68
Figura 13 – Estrutura <i>Sol_Ship</i> .....	69
Figura 14 - Pseudocódigo da meta-heurística proposta .....	70
Figura 15 – Representação da função de arranjo dos navios .....	72
Figura 16 – Representação da função de arranjo das pilhas .....	73
Figura 17 - Ilustração do Movimento 1 .....	74
Figura 18 – Ilustração do Movimento 2 .....	74
Figura 19 – Comparação entre o arranjo das pilhas em função do atributo <i>direcao</i> .....	75
Figura 20 – Pseudocódigo da função <i>CreateSolution</i> .....	77
Figura 21 – Pseudocódigo da função <i>Search_Position</i> .....	78
Figura 22 – Atualização do <i>skyline</i> em função da data planejada da pilha .....	78
Figura 23 - Pseudocódigo da função <i>Measure_Gap</i> .....	79
Figura 24 - Pseudocódigo da função <i>Temp_Block_Yard</i> .....	79
Figura 25 - Pseudocódigo da função <i>Update_Yard_Utilization</i> .....	80
Figura 26 - Pseudocódigo da função <i>Progress_Search</i> .....	81
Figura 27 - Pseudocódigo da função <i>Arrange_Ship</i> .....	82
Figura 28 – Resultado dos testes de calibração da temperatura inicial .....	85
Figura 29 – Resultado dos testes de calibração do coeficiente de resfriamento.....	86

Figura 30 – Resultado dos testes de calibração do número de iterações a cada temperatura .....	87
Figura 31 – Diferenças percentuais da FO em cada categoria de instâncias.....	91
Figura 32 – Diferenças percentuais de tempo de execução em cada categoria de instâncias .....	92
Figura 33 – Proporção de soluções viáveis e aprimoradas pelos movimentos na Instância 4.....	93
Figura 34 – Proporção de soluções viáveis e aprimoradas pelos movimentos na Instância 11 .....	94
Figura 35 – Proporção de soluções viáveis e aprimoradas pelos movimentos na Instância 16.....	94
Figura 36 – Convergência da FO ao longo da meta-heurística na Instância 4.....	95
Figura 37 – Convergência da FO ao longo da meta-heurística na Instância 11 .....	96
Figura 38 – Convergência da FO ao longo da meta-heurística na Instância 16.....	96

## LISTA DE TABELAS

Tabela 1 – Classificação dos trabalhos similares à corrente pesquisa .....	40
Tabela 2 – Instâncias utilizadas .....	60
Tabela 3 – Resultados obtidos pela meta-heurística e pelo CPLEX nas instâncias testadas.....	89

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>11</b>
1.1	OBJETIVOS .....	13
1.1.1	<b>Objetivo geral.....</b>	<b>13</b>
1.1.2	<b>Objetivos específicos.....</b>	<b>14</b>
1.2	JUSTIFICATIVA.....	14
1.3	ESTRUTURA DA PROPOSTA DE DISSERTAÇÃO .....	15
<b>2</b>	<b>REFERENCIAL TEÓRICO .....</b>	<b>17</b>
2.1	PROBLEMA DE ALOCAÇÃO DE BERÇOS .....	17
2.2	REVISÃO DA LITERATURA .....	20
2.3	MODELO MATEMÁTICO PROPOSTO POR BORDAL (2020) .....	40
<b>3</b>	<b>MÉTODO DE PESQUISA.....</b>	<b>52</b>
<b>4</b>	<b>PROBLEMA ESTUDADO E INSTÂNCIAS UTILIZADAS.....</b>	<b>55</b>
4.1	PROBLEMA ESTUDADO .....	55
4.2	INSTÂNCIAS UTILIZADAS .....	58
<b>5</b>	<b>DESENVOLVIMENTO DA META-HEURÍSTICA.....</b>	<b>62</b>
5.1	REPRESENTAÇÃO DOS PÁTIOS E DO CAIS.....	63
5.2	PARÂMETROS, ESTRUTURAS E VARIÁVEIS DESENVOLVIDAS .....	65
5.3	VISÃO GERAL DA META-HEURÍSTICA.....	69
5.4	DETALHAMENTO DAS FUNÇÕES .....	76
<b>6</b>	<b>RESULTADOS E ANÁLISES.....</b>	<b>84</b>
6.1	CALIBRAÇÃO DOS PARÂMETROS.....	84
6.2	RESULTADOS E ANÁLISES.....	88
<b>7</b>	<b>CONCLUSÕES .....</b>	<b>98</b>
	<b>REFERÊNCIAS.....</b>	<b>100</b>

# 1 INTRODUÇÃO

O minério de ferro é uma das commodities comercializadas de maior importância para a economia brasileira. Este item, insumo essencial na cadeia produtiva de diversos itens que utilizam o aço como matéria prima, representou uma participação 64,2% das exportações da indústria extrativa mineral nacional no primeiro semestre de 2018, contribuindo com cerca de US\$ 9 bilhões (DNPM, 2018).

A produção brasileira de minério de ferro possui grande representatividade no cenário internacional. No ano de 2017, o Brasil produziu cerca de 440 milhões de toneladas, o equivalente a 18% da produção mundial, valores que o colocam como o segundo maior produtor, ficando atrás apenas da Austrália, país responsável pela produção de 880 milhões de toneladas, quase 37% da produção total de minério de ferro (USGS, 2018).

Um dos principais insumos da indústria siderúrgica é a pelota de minério de ferro, um aglomerado de partículas finas de ferro obtido por meio do processo de pelotização. A pelotização do minério de ferro é um processo que surgiu com o intuito inicial de aproveitar os finos de minério gerado durante as etapas de lavra e beneficiamento e que não eram posteriormente utilizados (LUZ; SAMPAIO; FRANÇA, 2010). Em função de suas propriedades físico-químicas, as pelotas são utilizadas principalmente na alimentação dos altos-fornos, resultando em uma maior eficiência no processo de produção do aço (SAMARCO, 2019).

As pelotas de minério de ferro, produzidas em lotes de modo contínuo e ininterrupto, são estocadas em áreas de pátio de produto acabado e, posteriormente, escoadas por meio dos modos ferroviário e/ou aquaviário. Caso não haja disponibilidade no pátio, a produção dos lotes não pode ser iniciada, impactando financeiramente a empresa produtora. No que tange ao escoamento via modo aquaviário, visando à continuidade do processo produtivo, o planejamento das chegadas dos navios deve ser compatível com a disponibilização de áreas do pátio para a produção dos lotes (BORDAL, 2020).

Em função dos pontos apresentados, os planejadores das pelotizadoras enfrentam o seguinte problema: dada uma quantidade de lotes a serem produzidos em função dos contratos comerciais e um conjunto de navios, com datas estimadas de chegada (ETA) conhecidas, que virão ao porto para carregar os lotes, deve-se definir a alocação de espaço no pátio para as pilhas de pelotas, o momento em que cada pilha deverá ser formada e a sequência de atracação dos navios, visando à manutenção do ritmo de produção. Este problema é abordado no trabalho de Bordal (2020), que propôs um modelo matemático para o planejamento da fila de navios integrado à designação de áreas de pátio de armazenagem de carga a granel para atender ao plano de produção da usina de pelotização.

O processo decisório de definição da sequência de atracação dos navios em um porto é problema amplamente difundido na literatura científica. Considerando um conjunto de navios que devem ser atendidos em um determinado horizonte de tempo e as restrições físicas, operacionais e comerciais cabíveis ao sistema em estudo, o Problema de Alocação de Berços (PAB) consiste na designação do tempo de serviço e espaço no cais para o carregamento e descarregamento dos navios, usualmente com foco na redução do tempo de estadia do navio (BIERWIRTH e MEISEL, 2010).

Mesmo com a vasta quantidade de estudos publicados acerca do PAB, há uma escassez de trabalhos que o tratam em conjunto com o problema de designação de áreas de pátio para armazenagem, denominado *Yard Assignment Problem* (YAP), especialmente em terminais de carga a granel, uma questão posteriormente reforçada no referencial teórico deste trabalho.

Robenek *et al.* (2014) e Al-Hammadi e Diabat (2017) propuseram formulações para o problema em terminais multi-graneleiros com diversos pátios, permitindo designar qual tipo de carga abasteceria cada um dos pátios ao longo do horizonte de planejamento adotado. Boland, Gulczynski e Savelsbergh (2012), Babu *et al.* (2015) e Unsal e Oguz (2019) trataram do planejamento integrado em terminais de carvão, permitindo designar o momento em que cada uma das pilhas de carvão deveria ser formada e qual posição elas ocupariam nos pátios.

Apesar de considerarem o posicionamento das cargas nos pátios de armazenagem, os trabalhos supracitados tinham como foco a fila de navios, visando exclusivamente

minimizar o tempo de estádio dos navios no porto. Nenhum destes trabalhos promoveu a integração entre o PAB e o YAP para o atendimento do plano de produção de uma usina de pelotização, tendo como objetivo minimizar o atraso da produção das pilhas de pelota em relação às datas planejadas. Este problema foi formulado por Bordal (2020). Para a avaliação de seu modelo, desenvolvido no *solver* CPLEX 12.8, o autor em questão baseia-se em dados do Porto de Ubu, pertencente à usina de pelotização da empresa Samarco Mineração S/A.

O PAB pertence à classe dos problemas *NP-Hard* (LIM, 1998). Este ponto denota a complexidade deste problema, resultando em um enorme esforço computacional necessário para a resolução de instâncias de grande porte por meio de métodos exatos. Por esta questão, a proposta de métodos alternativos capazes de resolver o referido problema em menor tempo computacional, mantendo a qualidade da solução encontrada pelo modelo matemático, torna-se válida e justificável.

Diante dos pontos apresentados, esta pesquisa propôs o desenvolvimento de uma meta-heurística para a resolução do problema de planejamento integrado da designação das áreas de pátio para a armazenagem de lotes de pelotas de minério de ferro em conjunto com a definição da sequência de atracação dos navios ao longo do cais, visando atender ao plano de produção de uma usina de pelotização. Para a avaliação da meta-heurística desenvolvida, foram executadas as instâncias formuladas por Bordal (2020) e, posteriormente, os resultados obtidos foram comparados aos encontrados pelo modelo matemático, analisando a qualidade das soluções e o tempo necessário para a resolução do problema.

## 1.1 OBJETIVOS

### 1.1.1 Objetivo geral

Desenvolver uma meta-heurística aplicada ao processo de planejamento integrado da fila de navios e da alocação de áreas do pátio de armazenagem em um porto de carga a granel, visando atender o plano de produção de uma usina de pelotização.

### 1.1.2 Objetivos específicos

Os objetivos específicos da dissertação são:

- Realizar uma revisão bibliográfica de estudos relevantes sobre o problema de alocação de berços;
- Desenvolver a meta-heurística para a resolução do problema abordado;
- Definir adequadamente os parâmetros da meta-heurística proposta, por meio de um processo de calibração;
- Aplicar a meta-heurística ao problema real da usina de pelotização e do Porto de Ubu pertencentes à Samarco Mineração S/A;
- Analisar e comparar os resultados obtidos pela meta-heurística com os resultados alcançados por Bordal (2020).

## 1.2 JUSTIFICATIVA

Conforme discutido previamente, o Problema de Alocação de Berços (PAB) é um problema de otimização amplamente difundido na literatura científica, com diversos trabalhos publicados, como é possível observar nas extensivas revisões apresentadas em Bierwirth e Meisel (2010) e Bierwirth e Meisel (2015). Tal ponto é justificado visto que o berço é um recurso crítico na determinação da capacidade de um terminal portuário (PARK; KIM, 2003).

Apesar da sua contribuição para a eficiência das operações portuárias, a resolução do PAB de maneira isolada, entretanto, representa apenas um subproblema dentro do contexto logístico. Segundo Bierwirth e Meisel (2015), a velocidade na execução deste tipo de operação requer não somente planos eficientes para os berços, mas também outros fatores como o uso eficiente das áreas de armazenagem. Assim, a integração do PAB com o planejamento do pátio é motivada.

No que tange especificamente ao contexto das empresas pelotizadoras, Bordal (2020) destaca os impactos financeiros associados à paralisação da produção e às multas e quebras de contratos em função do não atendimento a demanda acordada. Assim, o

desenvolvimento de estudos que contribuam para o processo decisório de modo integrado neste contexto torna-se relevante.

Por fim, conforme comprovado por Lim (1998), o PAB, mesmo sem a integração com o planejamento das operações do pátio, é pertencente à classe dos problemas *NP-Hard*. Bierwirth e Meisel (2015) destacam que a maioria absoluta dos artigos revisados apresenta abordagens heurísticas como seus métodos mais bem sucedidos. Os autores destacam ainda que a riqueza dos modelos que tratam do PAB favorece a utilização de meta-heurísticas, permitindo abordar as diversas características associadas ao problema de modo flexível. Diante de tais pontos, é embasada a proposta de desenvolvimento de uma meta-heurística para a resolução do problema supracitado.

### 1.3 ESTRUTURA DA PROPOSTA DE DISSERTAÇÃO

O trabalho está estruturado em sete capítulos, incluindo o presente, que constitui a introdução da dissertação, apresentando a motivação, os objetivos e a justificativa que embasam esta pesquisa.

O Capítulo 2 apresenta o referencial teórico sobre os conceitos relativos ao problema abordado, descrevendo o Problema de Alocação de Berços (PAB), realizando a revisão de artigos sobre o assunto e apresentando o modelo matemático proposto por Bordal (2020), base para a meta-heurística desenvolvida neste estudo.

O Capítulo 3 exhibe o método de pesquisa, evidenciando as etapas que foram desenvolvidas no decorrer do trabalho.

O Capítulo 4 descreve o problema estudado, tendo como base a operação do Porto de Ubu, em Anchieta. Posteriormente, são apresentadas as instâncias utilizadas nesta pesquisa.

O Capítulo 5 descreve a meta-heurística desenvolvida para resolver o planejamento integrado da fila de navios e da alocação de áreas do pátio de armazenagem. A descrição apresenta os parâmetros, estruturas e variáveis utilizados pela meta-

heurística, a visão geral de seu funcionamento e as principais funções utilizadas para gerar as soluções do problema.

O Capítulo 6 apresenta os experimentos realizados, descrevendo o processo de calibração dos parâmetros, exibindo os resultados obtidos pela meta-heurística nas instâncias utilizadas, comparando esses resultados com os obtidos pelo modelo matemático da literatura e, por fim, analisando o comportamento da meta-heurística.

Por fim, o Capítulo 7 apresenta as considerações finais do trabalho, concluindo a pesquisa e indicando sugestões de trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Neste capítulo é feito o embasamento teórico deste estudo. Na primeira seção é feita uma descrição do Problema de Alocação de Berços (PAB), foco deste estudo. Na segunda seção é feita uma revisão bibliográfica de diversos artigos da literatura internacional acerca do PAB. Por fim, é apresentado o modelo matemático proposto Bordal (2020), base para o desenvolvimento do presente trabalho.

### 2.1 PROBLEMA DE ALOCAÇÃO DE BERÇOS

Um dos problemas em destaque dentro do contexto do planejamento das operações portuárias é Problema de Alocação de Berços (PAB), que, como destacado previamente, trata a alocação do tempo de serviço e de espaços no cais para o carregamento e descarregamento de navios nos terminais portuários (BIERWIRTH; MEISEL, 2010).

Considerando um determinado layout do cais e um conjunto de navios a serem atendidos dentro de certo horizonte de tempo, o PAB consiste em um problema que visa à otimização de um determinado objetivo, usualmente associado à minimização dos custos ou do tempo total de serviço, a partir das decisões de atribuição de tempos e posições de atracação para os navios (BIERWIRTH; MEISEL, 2015; CORDEAU *et al.*, 2005).

De modo genérico, o PAB tem por objetivo prover um melhor atendimento aos navios que entram no sistema, principalmente em termos de rapidez e confiabilidade. Tal ponto baseia-se no interesse das partes envolvidas, tendo em vista a perda de rentabilidade associada a uma longa retenção dos navios nas filas.

Na literatura científica, modelos matemáticos com os mais diversos objetivos são encontrados, tais como: minimização do tempo de estadia do navio, minimização da carga de trabalho dos recursos e, minimização do número de navios não atendidos no terminal (BIERWIRTH; MEISEL, 2010).

Considerando as métricas de desempenho adotadas, destaca-se que diversos objetivos possuem uma relação de dependência entre si e com outros fatores. Considerando o exemplo supracitado sobre a minimização do tempo de estadia, este pode ser alcançado, por exemplo, por meio dos seguintes objetivos: minimização do tempo de espera para atracação, minimização do tempo de atendimento aos navios e minimização de saídas atrasadas (BIERWIRTH; MEISEL, 2015).

Um modo comumente utilizado para classificar as formulações do PAB diz respeito aos atributos temporais e espaciais. No que tange o aspecto temporal, o principal ponto analisado é o processo de chegada dos navios, classificado em: 1) chegada estática, 2) chegada dinâmica, 3) chegada cíclica e 4) chegada estocástica (BIERWIRTH; MEISEL, 2015).

Os modelos baseados em chegada estática consideram que todos os navios já estão aguardando na área de fundeio do porto, podendo atracar imediatamente, assim que forem liberados. No caso dos modelos com chegada dinâmica, trabalha-se com horários de chegada fixos e bem definidos, impedindo que um navio seja liberado para atracar antes do momento de chegada ao porto. Usualmente, toma-se como base o momento estimado de chegada do navio, denominado *Estimated Time of Arrival* (ETA).

Os modelos que adotam um regime de chegada cíclica consideram um horizonte de planejamento longo, onde cada navio possui uma programação definida, usualmente estabelecida em contrato. Assim, estes navios chegam ao porto repetidamente em intervalos de tempo fixos pré-estabelecidos.

Nos modelos com chegadas estocásticas, assume-se um cenário de incertezas onde a chegada do navio pode ser influenciada por fatores alheios ao escopo do sistema portuário tratado. Neste caso, os momentos de chegada são determinados por distribuições contínuas aleatórias ou por cenários com probabilidades discretas de ocorrência.

A taxonomia que classifica o PAB conforme atributos espaciais diz respeito, principalmente, à forma como o cais é dividido. Baseado nela, o PAB pode ser classificado como: 1) PAB Discreto (PABD), 2) PAB Contínuo (PABC) e 3) PAB Híbrido

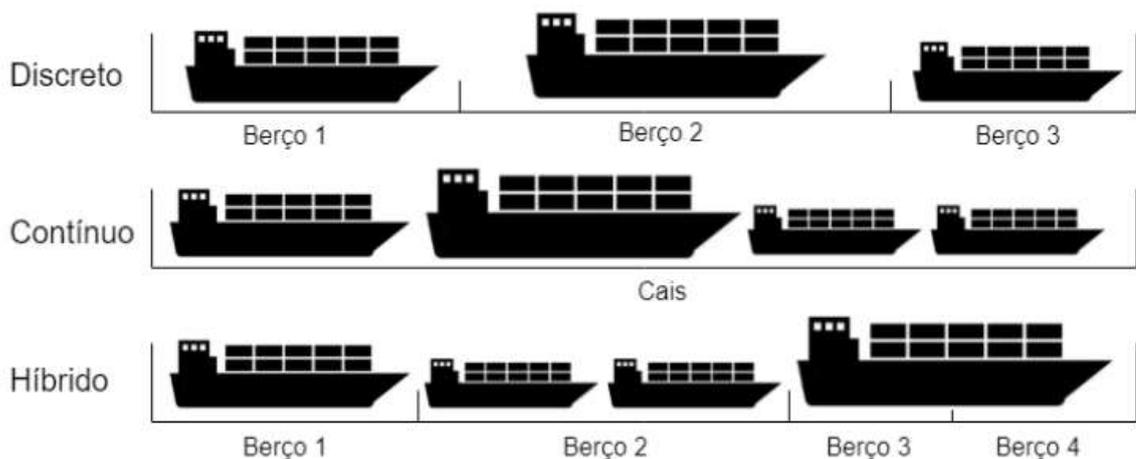
(PABH) (IMAI *et al.*, 2005). A Figura 1 representa graficamente as variações apresentadas do PAB.

No PABD, o cais é dividido em seções (berços), de modo que cada berço só pode atender um único navio de cada vez. Este particionamento do cais, que também é necessário ao PABH, pode acompanhar a construção do cais ou ser prescrito para facilitar o processo de planejamento.

O PABC configura a situação em que não há divisão do cais, permitindo que os navios atracuem em posições arbitrárias, desde que estejam dentro dos limites do cais. Tal modelo permite uma melhor utilização do cais. Entretanto, nesta situação de cais contínuo, o planejamento torna-se bem mais complexo do que no caso com berços discretos.

O PABH é um caso que se assemelha ao PABD por promover a divisão do cais em berços, porém permitindo certa flexibilidade. Nesta configuração, embarcações maiores podem ocupar mais de um berço, enquanto navios menores possuem a possibilidade de compartilhar uma única vaga.

Figura 1 – Representação gráfica do PAB discreto, contínuo e híbrido



Fonte: Adaptado de Bierwirth e Meisel (2010).

Outras questões podem ser consideradas para uma programação adequada dos berços. Fatores como a relação entre as janelas de tempo dos berços e das embarcações, a quantidade e o tipo de carga a ser carregada ou descarregada no porto, profundidade do cais, variação da maré, entre outros, influenciam diretamente nas restrições temporais e espaciais do problema (LALLA-RUIZ *et al.*, 2016).

No próximo subcapítulo, é feita de uma revisão bibliográfica dos estudos levantados acerca do PAB.

## 2.2 REVISÃO DA LITERATURA

Imai, Nishimura e Papadimitriou (2001) apresentaram uma formulação para a versão dinâmica do problema de alocação de berços discretos (PABD-D), considerando terminais de contêineres em um sistema público do Japão, tendo como objetivo minimizar a somatória dos tempos de espera e manuseio dos navios. Neste modelo, os tempos de manuseio de cada navio são heterogêneos, variando em função da posição de atracação do navio.

Dada a complexidade do problema, os autores propuseram uma heurística baseada em relaxação lagrangeana (RL) como método de solução. O método proposto foi testado em instâncias geradas com bases em dados obtidos no porto de Kobe, validando a heurística, julgada pelos autores como adaptável para problemas de dimensões práticas.

Nishimura, Imai e Papadimitriou (2001) continuaram estenderam a sua pesquisa, considerando a possibilidade de atendimentos simultâneos de navios em um mesmo berço, configurando um problema com berços híbridos. O modelo proposto também visa a minimização dos tempos de espera e atendimento, mas diferencia-se, principalmente, pelas restrições físicas consideradas. Nesta formulação, cada berço possui diferentes dimensões de comprimento e profundidade, de modo a limitar a atracação dos navios em função de suas dimensões de calado e comprimento total (LOA).

Os autores desenvolveram uma heurística baseada em algoritmos genéticos (GA) para a resolução do problema. Experimentos computacionais foram promovidos comparando os resultados obtidos pelo GA com o método de RL desenvolvido em Imai, Nishimura e Papadimitriou (2001), considerando os berços como discretos. Os resultados obtidos demonstraram a superioridade da relaxação lagrangeana em termos de qualidade da solução, principalmente em instâncias de maior porte.

Posteriormente, o método desenvolvido foi testado considerando atendimentos simultâneos nos berços, sendo comparado aos dados obtidos a partir de casos reais. Os resultados obtidos demonstraram a aplicabilidade dos algoritmos genéticos, gerando planos capazes de promover um aumento na eficiência da utilização dos berços.

Imai, Nishimura e Papadimitriou (2003) se basearam na formulação apresentada em Imai, Nishimura e Papadimitriou (2001) e propuseram um modelo para o PABD-D, considerando diferentes níveis de prioridade de atendimento para cada navio. Assim, o modelo tem por objetivo minimizar o tempo de total de serviço, composto pela soma dos tempos de espera e atendimento, multiplicado pelo peso atribuído aos navios em função das prioridades previamente definidas.

Os autores consideraram inicialmente a resolução do problema por meio de uma heurística baseada em RL, mas o problema relaxado resultante desta abordagem também era *NP-hard*, com enorme esforço computacional esperado para sua resolução. Em função do sucesso obtido na aplicação do GA por Nishimura, Imai e Papadimitriou (2001), esta abordagem foi utilizada pelos autores, sendo validada por meio dos testes computacionais em instâncias baseadas em dados reais do porto de Kobe.

Imai *et al.* (2005) continuaram as pesquisas tomando como base o PAB dinâmico com cais contínuo (PABC-D), também com o objetivo de minimizar o tempo total de serviço. Como método de solução, os autores propuseram uma heurística que obtém soluções para o problema com berços discretos e posteriormente reposiciona os navios respeitando as características inerentes à continuidade do cais.

Cordeau *et al.* (2005) abordaram duas versões do PAB dinâmico, uma considerando berços discretos e outra com cais contínuo. O objetivo em ambos os casos é minimizar a soma ponderada dos tempos de serviço, composta pelos tempos de espera e atendimento de cada navio multiplicado pelo valor associado ao tempo de serviço.

Para tratar do caso discreto, Cordeau *et al.* (2005) propuseram uma nova formulação baseada no problema de roteamento de veículos multi-depósito com janela de tempo (PRVMD-JT) e desenvolveram uma meta-heurística Tabu Search (TS) como método de solução. A técnica proposta foi testada em instâncias desenvolvidas a partir de

dados coletados do porto de Gioia Tauro e comparada com os resultados obtidos pelo modelo resolvido em *solver* comercial. Os resultados demonstraram a capacidade da meta-heurística encontrar a solução ótima em instâncias de pequeno porte e obter soluções melhores do que o *solver* dentro dos limites de tempo estabelecidos em instâncias maiores.

No que tange o caso contínuo, os autores também desenvolveram uma meta-heurística TS, comparando-a em experimentos computacionais com uma heurística construtiva semelhante ao processo de planejamento vigente à época. Os resultados obtidos demonstraram a capacidade do método proposto de obter melhores soluções.

Imai *et al.* (2007) apresentaram uma nova formulação por programação linear inteira mista (MILP) para o PAB dinâmico com berços híbridos (PABH-D), em contraposição as formulações não lineares apresentadas anteriormente pelos autores. O modelo tem por objetivo a minimização dos tempos totais de espera e atendimento.

Dois modelos alternativos também foram apresentados nesta pesquisa. O primeiro modelo considera prioridades para os denominados mega navios, visando garantir o atendimento destes no momento de chegada ao terminal, dado os altos custos inerentes aos atrasos. O segundo modelo considera uma configuração do terminal com berços indentados, nos quais os mega navios deverão ser atendidos. Este layout implica em novas limitações físicas para atracação e desatracação de navios.

Para a resolução das formulações apresentadas, os autores propuseram a utilização de algoritmos genéticos (GA). Os experimentos computacionais conduzidos indicaram que o terminal com berços indentados apresenta menor eficiência global em sua operação, apesar de melhorar individualmente os tempos de atendimento dos mega navios.

Hansen, Oguz e Mladenovic (2008) propuseram uma nova formulação MILP para o PABC-D, visando a minimização dos custos totais. Para cada navio, são atribuídos diferentes custos relacionados à sua espera e ao seu atendimento em cada um dos berços. Além disso, são definidas datas esperadas para o término dos serviços, que usualmente são especificadas em contrato. A conclusão de um serviço antes do período de tempo estabelecido implica em um prêmio para a empresa que opera o porto, enquanto um atraso implica em uma penalidade.

A resolução do problema por *solver* comercial só foi aplicável em instâncias de pequeno porte, dada a complexidade do problema. Assim, os autores desenvolveram uma meta-heurística *Variable Neighborhood Search* (VNS), comparando-a posteriormente com os métodos Multi-Start (MS), GA e Algoritmos Meméticos (MA). Os experimentos indicaram a superioridade do algoritmo VNS ao obter melhores resultados em todas as instâncias testadas.

Imai, Nishimura e Papadimitriou (2008) apresentaram duas novas formulações, considerando chegadas estáticas e dinâmicas dos navios, para o PABD. Neste estudo, os autores incluíram a possibilidade de não atendimento dos navios, dados os limites de tempo estabelecidos para a execução dos serviços. Os navios que não forem atendidos no terminal desejado são destinados a um terminal externo para a conclusão de suas operações.

As formulações apresentadas tem por objetivo a minimização do tempo total de atendimento executado no terminal externo. No que tange a resolução do problema, os autores focaram na resolução do problema dinâmico, desenvolvendo uma heurística baseada em GA.

Imai *et al.* (2008) promoveram a integração do PAB com o problema de alocação de guindastes (QCAP), apresentando um modelo para a resolução simultânea de ambos os problemas. Na formulação apresentada, com chegadas dinâmicas e berços discretos, são determinadas quantidades específicas de guindastes necessários para a operação de cada navio. Deste modo, as operações só se iniciam com a disponibilidade prévia dos guindastes.

O modelo proposto tem por objetivo minimizar os tempos de espera e atendimento. Destaca-se que, considerando as premissas impostas na formulação, a velocidade de atendimento não é explicitamente influenciada pela quantidade de guindastes que o servem, representando uma limitação do modelo. Como método de solução, os autores propuseram a utilização de algoritmos genéticos, destacando, entretanto, que a qualidade das soluções não foi examinada nesta pesquisa.

Liang, Huang e Yang (2009) também abordaram simultaneamente o PAB e o QCAP, considerando chegadas dinâmicas e berços discretos. O modelo proposto visa

minimizar os tempos totais de espera e atendimento, além dos tempos de atraso em função dos prazos estabelecidos.

Diferentemente da premissa considerada em Imai *et al.* (2008), os autores não impuseram uma quantidade fixa de guindastes para cada navio, apenas um limite máximo que pode ser empregado em cada atendimento. Assim, diferentes quantidades podem ser alocadas, de modo que o tempo necessário para a conclusão do serviço é associado diretamente ao número de guindastes em operação. Para a resolução deste problema, os autores empregaram a utilização de algoritmos genéticos.

Barros *et al.* (2011) abordaram o PABC-D em um terminal de carga a granel, apresentando uma nova formulação MILP. Neste trabalho, destaca-se a inclusão de restrições para atracação e desatracação em função das variações de maré, limitando tais movimentações as janelas de tempo de maré alta.

Outro destaque do modelo tange as restrições associadas aos níveis de estoque dos diferentes tipos de granéis considerados. Deste modo, um navio destinado a carregar um determinado tipo de carga só será atendido caso haja disponibilidade em estoque para tal. Ao visar a minimização dos custos totais de serviço, incluindo o demurrage, custo associado ao atraso no atendimento em relação à data estabelecida e contrato, o modelo acaba priorizando o atendimento de navios destinados ao descarregamento de granéis com níveis críticos de estoque.

Para a resolução do modelo, os autores desenvolveram uma meta-heurística baseada em *Simulated Annealing* (SA). Os experimentos computacionais foram promovidos em instâncias geradas com base em dados reais da operação do Terminal de São Luís. Os resultados obtidos pelo SA foram comparados com a execução em *solver* comercial, alcançando a solução ótima na maioria das instâncias com um esforço computacional significativamente menor.

Zhen, Lee e Chew (2011) resolveram o PABC considerando um cenário de incertezas, aproximando-se mais da realidade da operação portuária. Nesta abordagem os tempos de chegada e de operação dos navios são considerados estocásticos. Esta incerteza é representada por meio de um conjunto finito de cenários discretos.

Os autores desenvolveram um modelo decisório de duas etapas, incorporando tanto uma estratégia proativa de construção de um plano inicial quanto uma estratégia reativa de adequar o planejamento ao longo do horizonte de tempo. O modelo visa minimizar os custos de espera e os custos de atendimento, associados também a posição de atracação, além de também estabelecer um custo associado à modificação do planejamento previamente estabelecido. Como método de resolução, os autores desenvolveram um algoritmo baseado na meta-heurística SA, capaz de encontrar soluções ótimas ou próximas do valor ótimo nas instâncias testadas.

Zhen, Chew e Lee (2011) formularam um novo modelo para integrar o PABC ao QCAP e ao problema de designação de áreas de pátio para armazenagem da carga, referido aqui como *Yard Assignment Problem* (YAP). A abordagem proposta teve como foco um hub de transbordo de contêineres, com um regime de chegadas cíclicas por parte dos navios.

Os autores desenvolveram um modelo de programação linear inteira mista (MILP) cujo os objetivos eram minimizar os custos associados ao atraso do atendimento aos navios e às movimentações dos contêineres do cais ao pátio, variáveis em função do distanciamento da posição de atracação dos navios em relação à posição estabelecida para armazenagem de suas cargas. Para resolver instâncias fidedignas à realidade da operação, eles desenvolveram um algoritmo baseado na meta-heurística *Squeaky Wheel Optimization* (SWO), encontrando boas soluções em um tempo computacional apropriado.

Boland, Gulczynski e Savelsbergh (2012) abordaram de modo inédito o planejamento integrado em um terminal de exportação de carvão em Newcastle, na Austrália. A abordagem proposta trata conjuntamente o PABD, o YAP e a programação de empilhadeiras e recuperadoras nos pátios de estocagem de carvão, responsáveis pela formação das pilhas no pátio e a recuperação destas para os carregadores de navio, respectivamente. Para resolver este problema integrado, tendo como objetivo minimizar o atraso na atracação dos navios, os autores propuseram um algoritmo híbrido, combinando heurísticas construtivas gulosas com enumeração e programação inteira.

Zhen e Chang (2012) apresentaram uma nova abordagem para o PABC sob condições de incertezas, propondo um modelo bi-objetivo que visa minimizar os custos, referentes ao atraso no atendimento e a atracação do navio fora da posição adequada, ao mesmo tempo em que maximiza a robustez do plano elaborado face às incertezas.

Diferentemente do modelo decisório apresentado em Zhen, Lee e Chew (2011), os autores empregaram apenas a estratégia proativa, desconsiderando a adaptação do plano ao longo do horizonte de planejamento. Outra diferença notável corresponde ao fato do modelo ser bi-objetivo, encontrando, como resultado, um conjunto de soluções não dominadas ao invés de uma única solução, permitindo, assim, analisar o *tradeoff* entre custos e robustez. Para a resolução do problema, os autores desenvolveram uma heurística dedicada ao problema, capaz de encontrar soluções em problemas de larga escala.

Oliveira, Mauri e Lorena (2012) abordaram o PABC-D tendo como objetivo minimizar a soma ponderada dos tempos de espera e atendimento dos navios, considerando restrições de janela de tempo. Como método de solução, os autores utilizaram uma abordagem híbrida denominada *Clustering Search* (CS), um método iterativo que localiza regiões promissoras dentro do espaço de busca, utilizando uma meta-heurística SA embutida para gerar soluções.

O método desenvolvido foi testado em instâncias geradas com bases nos dados de Cordeau *et al.* (2005), sendo comparado com outras meta-heurísticas da literatura e com o modelo executado em *solver* comercial. A técnica proposta foi capaz de encontrar as melhores soluções conhecidas, até então, com um menor tempo de execução em comparação aos demais métodos.

Xu, Li e Leung (2012) abordaram as versões dinâmica e estática do PABD, considerando a variação da maré e diferentes profundidades de berços em seu modelo MILP. Tais aspectos implicam na restrição de atracação e desatracação aos períodos de maré alta. Além disso, certos berços só podem atender navios de calados menores em função de sua profundidade. O objetivo do modelo é minimizar a soma ponderada dos tempos de atendimento.

Visto a complexidade do problema, os autores desenvolveram duas heurísticas dedicadas, uma para o cenário com chegadas estáticas e outro para chegadas dinâmicas. As heurísticas foram comparadas com os *lower bounds* obtidos pela execução do modelo em *solver* comercial, obtendo resultados que inferem, segundo os autores, a eficácia dos métodos propostos.

Lee e Jin (2013) formularam um modelo MILP para integrar o PABD ao YAP, considerando um regime de chegadas cíclicas em um terminal de transbordo de contêineres. O objetivo do modelo era minimizar os custos relativos ao fluxo dos contêineres no porto, bem como minimizar o desbalanço da carga de trabalho realizada ao longo do horizonte de planejamento. Em conjunto com o modelo, foi desenvolvida uma meta-heurística baseada em algoritmos meméticos (MA) para resolver instâncias de médio e grande porte.

Hendriks, Lefebvre e Udding (2013) abordaram o PABC e o YAP de modo integrado, também considerando um regime de chegadas cíclicas em um terminal de transbordo de contêineres. Os autores formularam um modelo de programação quadrática inteira mista (MIQP) com o objetivo de minimizar os custos associados ao fluxo de contêineres. Uma abordagem heurística foi desenvolvida para a resolução do problema. O método, que alterna de modo iterativo entre planejar a atracação dos navios e designar as posições das cargas no pátio, convergia rapidamente para mínimos locais do espaço de busca das soluções.

Umang, Bierlaire e Vacca (2013) abordaram o PABH-D visando minimizar os tempos totais de espera e atendimento nas operações de terminais de carga a granel. Os tipos de carga foram explicitamente considerados no estudo e os tempos de atendimento estão diretamente atrelados à quantidade a ser carregada ou descarregada e a distância da posição de atracação do navio em relação à posição do tipo de carga no pátio. As seções ao longo do cais apresentam diferentes profundidades, limitando a atracação dos navios em função das dimensões de calado.

Os autores apresentaram duas abordagens exatas para o problema: um modelo de programação linear inteira mista (MILP) e uma formulação denominada *Generalized Set Partitioning Problem* (GSPP). Além disso, os autores desenvolveram um algoritmo

baseado na meta-heurística *Squeaky Wheel Optimization* (SWO) como método alternativo.

Para testar os métodos apresentados, instâncias de teste foram geradas com base em dados extraídos do porto SAQR, nos Emirados Árabes. Os resultados obtidos atestaram a superioridade da formulação por GSPP, capaz de obter as soluções ótimas no problema apresentado, com menor tempo computacional em relação ao modelo MIP. A heurística desenvolvida foi julgada válida para utilização em instâncias nas quais o GSPP demanda um esforço computacional excessivo.

Robenek *et al.* (2014) estenderam o trabalho de Umang, Bierlaire e Vacca (2013) integrando o PAB ao problema de designação das áreas do pátio para armazenagem das cargas, denominado *Yard Assignment Problem* (YAP). Diferentemente do trabalho anterior, onde as áreas do pátio para cada tipo de carga são previamente estabelecidas, a abordagem proposta permite designar quais as posições do pátio de armazenagem serão operadas, tanto para as operações de carregamento quanto descarregamento.

Para a resolução exata do problema, os autores empregaram uma abordagem denominada *Branch and Price* (B&P), combinando os métodos *Branch and Bound* (B&B) e Geração de Colunas (GC). De modo alternativo, foi proposta uma meta-heurística *Critical-Shaking Neighborhood Search* (CSNS), visando encontrar boas soluções em instâncias de maior porte com menor esforço computacional.

Turkogullari *et al.* (2014) consideraram a integração do PAB com o QCAP em terminais de contêineres sob um cenário de chegadas dinâmicas e berços híbridos. Dois modelos MILP foram apresentados. A primeira formulação considera apenas a alocação de guindastes em termos numéricos, enquanto a última leva em consideração a alocação específica de cada guindaste, identificado individualmente. O objetivo de ambas as formulações é minimizar os custos associadas a espera para atracação, ao atraso no término do serviço e a atracação em uma posição não desejada.

Os experimentos computacionais promovidos indicaram que o primeiro modelo pode ser resolvido à sua otimalidade em instâncias com até 60 navios, ao contrário do que ocorre com o segundo modelo. Tal fato justificou o desenvolvimento de um algoritmo

com base no método *Branch and Cut* (B&C), permitindo utilizar soluções ótimas do primeiro modelo para gerar soluções ótimas com especificação dos guindastes em tempo hábil.

Imai, Yamakawa e Huang (2014) destacaram a escassez de trabalhos que abordam o PAB sob o ponto de vista estratégico. Assim, os autores apresentam uma nova formulação por programação inteira adaptada com base no modelo desenvolvido em Imai, Nishimura e Papadimitriou (2001). Nesta nova abordagem, os navios seguem um regime de chegadas cíclicas com base em contratos de longo prazo, com tempos de operação iguais em cada data para fins de simplificação.

O modelo proposto permite a decisão estratégica de rejeitar um serviço, tendo como objetivo minimizar os custos associados aos tempos de espera, a ociosidade dos berços e as penalidades inerentes a rejeição dos serviços. Para a resolução de instâncias de grande porte, os autores desenvolveram uma heurística baseada em relaxação lagrangeana.

Hu, Hu e Du (2014) consideraram a integração do PAB com o QCAP, considerando chegadas dinâmicas e um cais contínuo, abordando questões ambientais relacionadas ao consumo de combustível e à emissão de poluentes. Foi desenvolvido um modelo não-linear multiobjetivo, linearizado posteriormente para viabilizar a sua resolução, promovendo o *tradeoff* entre a redução dos custos associados a operação e os fatores ambientais incluídos.

Os autores destacaram que, diferentemente de outros trabalhos encontrados, este estudo considera o impacto da alocação dos guindastes no consumo de combustível e nas emissões. Uma heurística dedicada foi desenvolvida como método de solução para o modelo apresentado.

Ting, Wu e Chou (2014) apresentaram uma formulação para o PABD-D baseando-se em uma formulação para PRV-MD-JT, semelhante à abordagem apresentada em Cordeau *et al.* (2005), visando minimizar os custos associados aos tempos de espera e atendimento dos navios. Os autores desenvolveram um algoritmo baseado na meta-heurística *Particle Swarm Optimization* (PSO), que, até então, não havia sido encontrada em trabalhos anteriores.

Para testar a eficiência do PSO, os autores promoveram experimentos utilizando as instâncias de benchmarking de Cordeau *et al.* (2005), comparando o seu algoritmo com diversos métodos apresentados na literatura incluindo a meta-heurística TS (CORDEAU *et al.*, 2005) e o método CS (OLIVEIRA; MAURI; LORENA, 2012). O algoritmo PSO foi capaz de encontrar a solução ótima em todas as instâncias com baixo tempo computacional, indicando a eficácia desta abordagem.

Hu (2015) abordou o PABD-D tendo como diferencial a inclusão da preferência pela realização dos atendimentos no período diurno. Tal ponto é justificado pela melhoria do conforto e segurança dos trabalhadores, além do aspecto sustentável inerente à economia de energia. O problema é formulado como um modelo de programação linear inteira mista (MILP) bi-objetivo, visando conjuntamente a minimização da carga de trabalho em atraso e a carga de trabalho realizada em período noturno.

Para resolver o problema de modo a permitir a análise do *tradeoff* entre os objetivos conflitantes, os autores desenvolveram um algoritmo genético multiobjetivo denominado *Non-dominated Sorting Genetic Algorithm II* (NSGA-II). O método utilizado permite encontrar boas aproximações da fronteira de Pareto, o conjunto de múltiplas soluções não dominadas por outras soluções factíveis.

Yan *et al.* (2015) consideraram o PAB dinâmico para propor um novo modelo, formulando-o como um modelo de fluxo de rede multi-commodities. No que tange ao arranjo dos berços, os autores denominaram a sua formulação como um problema de berços flexíveis, onde cada berço é representado por um conjunto finito de blocos. Tal abordagem, que se encaixa na classificação de berços híbridos, permitiu evitar algumas das desvantagens inerentes as formulações contínua e discreta. O modelo proposto, executado e validado por meio de *solver* comercial em instâncias de teste, é compatível com a regra *First Come First Serve* (FCFS), comum no contexto de atracação dos navios.

Babu *et al.* (2015) propuseram dois algoritmos baseados em heurísticas gulosas para resolver o planejamento integrado da operação de um terminal de importação de carvão localizado na Índia, visando minimizar o tempo de atraso ao atendimento dos navios. O problema tratado é uma integração entre o PABD, o YAP, a programação

das empilhadeiras/recuperadoras e a programação dos trens, que são carregados com as pilhas importadas.

Jin, Lee e Hu (2015) deram continuidade ao trabalho de Lee e Jin (2013), abordando conjuntamente o PABD e o YAP em um terminal de transbordo de contêineres, visando minimizar os custos relativos ao fluxo dos contêineres no porto e o desbalanço da carga de trabalho ao longo do horizonte de planejamento. Os autores desenvolveram uma heurística baseada no método de Geração de Colunas (GC) para resolver instâncias realistas, obtendo resultados próximos das soluções ótimas em tempo computacional aceitável para a sua aplicação na prática da operação portuária.

Tao e Lee (2015) propuseram um modelo MIQP para a resolução integrada do PABH com YAP em terminais de contêineres, também visando a minimização dos custos relativos ao fluxo dos contêineres no porto e ao desbalanço da carga de trabalho. Para a resolução do problema, os autores adotaram uma estratégia *multi-cluster*, dividindo os fluxos de transbordo de contêineres em grupos e aplicando uma estratégia baseada em *Tabu Search* (TS) para definir o arranjo dos contêineres e as posições de atracação dos navios associados a estes.

Liu *et al.* (2016) propuseram um modelo bi-objetivo para a integração do PABC com QCAP e YAP em terminal de transbordo de contêineres, tendo como objetivos minimizar os custos associados ao atraso do atendimento aos navios e às movimentações dos contêineres do cais ao pátio. Para encontrar soluções próximas à fronteira de Pareto, os autores desenvolveram uma meta-heurística baseada no NSGA-II.

deram continuidade ao trabalho de Lee e Jin (2013), abordando conjuntamente o PABD e o YAP em um terminal de transbordo de contêineres, visando minimizar os custos relativos ao fluxo dos contêineres no porto e o desbalanço da carga de trabalho ao longo do horizonte de planejamento. Os autores desenvolveram uma heurística baseada no método de Geração de Colunas (GC) para resolver instâncias realistas, obtendo resultados próximos das soluções ótimas em tempo computacional aceitável para a sua aplicação na prática da operação portuária.

Sahin e Kuvvetli (2016) abordaram o PABC-D tendo como objetivo a minimização dos custos totais, compostos pelo distanciamento da posição ideal de atracação e pelos

atrasos na conclusão dos serviços. Para a resolução do problema, os autores propuseram o método *Differential Evolution Algorithm* (DE), um algoritmo evolucionário, que consiste em um método estocástico de busca direta, desenvolvido para resolver funções espaciais contínuas.

Os autores desenvolveram aleatoriamente instâncias de teste para avaliar a eficiência do modelo, promovendo comparações com algumas meta-heurísticas da literatura e com os resultados obtidos pelo modelo MILP executado em *solver* comercial. Os resultados demonstraram a melhor convergência do método em comparação aos demais métodos, demonstrando o seu potencial.

Mauri *et al.* (2016) abordaram as versões contínua e discreta do PAB com chegadas dinâmicas, desenvolvendo uma meta-heurística *Adaptive Large Neighborhood Search* (ALNS) como método de solução, visando minimizar os tempos de permanência dos navios no porto. Como destaque pelos autores, este foi o primeiro trabalho a aplicar esta meta-heurística ao PAB.

Foram promovidos experimentos computacionais em três conjuntos de instâncias, comparando a ALNS com diversas abordagens promissoras da literatura. Para o caso discreto, a meta-heurística proposta foi capaz de encontrar todas as soluções ótimas conhecidas, além de 30 novas melhores soluções. Para o problema contínuo, todas as melhores soluções conhecidas do terceiro conjunto de instâncias foram encontradas, além de 10 novas melhores soluções para o primeiro conjunto. Tais resultados atestaram a superioridade do método desenvolvido.

Ribeiro *et al.* (2016) também empregaram a utilização da meta-heurística ALNS, aplicando-a como método de solução para nova formulação por programação linear inteira mista (MILP). O modelo desenvolvido visa representar o processo de alocação de berços em um terminal de minério de ferro em Vitória, composto por três berços discretos. A chegada dos navios é considerada como dinâmica nesta abordagem.

Um dos principais aspectos tratados pelo modelo é o *laytime*, um período pré-estabelecido em contrato no qual o navio deve ser atendido. Caso o *laytime* seja excedido, é incorrida uma multa (*demurrage*) à administração do porto. Entretanto, se o navio for atendido previamente ao *laytime*, o dono do navio paga uma recompensa (*despatch*) ao porto. Estes valores de *demurrage* e *despatch* são definidos em

contrato e podem ser diferentes para cada navio. O modelo tem como objetivo minimizar os custos associados a estas taxas.

Outro aspecto tratado diz respeito às atividades de manutenção nos berços, essenciais para reparar e preservar os equipamentos responsáveis pelo atendimento aos navios. Tal manutenção, representada por um navio “dummy” impede a atracação no berço durante a sua realização.

Experimentos computacionais foram promovidos em instâncias geradas com base em dados coletados do porto. O ALNS foi aplicado 10 vezes em cada instância, com um tempo computacional máximo de 120s, enquanto o modelo foi executado em *solver* comercial com um limite de 7200s. Os resultados indicaram que a meta-heurística foi capaz de obter as soluções ótimas nas 5 primeiras instâncias, além de melhorar as soluções encontradas pelo *solver* nos demais testes.

He (2016) integrou o PAB com o QCAP abordando o *tradeoff* entre economia de tempo e economia de energia. Neste modelo, o tempo de atendimento dos navios é influenciado diretamente pelo número de guindastes alocados e pela posição de atracação do navio. Cada navio tem uma melhor posição de atracação e um número máximo de guindastes que podem ser empregados para a sua operação.

O modelo MILP desenvolvido tem como objetivos a minimização dos atrasos totais de saídas do navio e a minimização dos custos associados ao consumo de energia pelos guindastes. Para a resolução do problema, os autores empregaram uma abordagem integrada de simulação e otimização, na qual a simulação é utilizada para avaliação, enquanto o algoritmo de otimização é responsável pela procura das soluções. Como método de otimização, os autores empregaram um Algoritmo Memético (MA), no qual são utilizados os métodos GA, para busca global, e SA, para busca local.

Kordic *et al.* (2016) propuseram uma abordagem exata inédita para as versões discreta e híbrida do PAB dinâmico, implementando-a em duas variantes. O método desenvolvido é denominado *Sedimentation Algorithm* (SEDA), um algoritmo de otimização combinatória que funciona com base nos princípios do *Branch and Bound* (B&B). A variante alternativa consiste em utilizar uma heurística como pré-processamento para reduzir o espaço de busca do método SEDA. Os experimentos computacionais promovidos indicaram a capacidade do algoritmo em encontrar as

soluções ótimas nas instâncias testadas, diminuindo significativamente o tempo computacional de execução.

Lalla-Ruiz *et al.* (2016) tomaram como base o modelo proposto por Xu, Li e Leung (2012) e desenvolveram uma nova formulação GSPP para PAB com diferentes profundidades de berço e condições de maré. Esta abordagem define múltiplos períodos de maré e inclui novas restrições, garantindo a viabilidade das soluções encontradas em face ao modelo de Xu, Li e Leung (2012). O modelo, que visa à minimização dos tempos totais de serviço, também inclui restrições de janela de tempo para o atendimento dos navios.

Para testar a formulação proposta, os autores utilizaram um conjunto de instâncias geradas por Xu, Li e Leung (2012), além de gerar novas instâncias. A nova formulação foi capaz de encontrar as soluções com menor tempo de execução, indicando uma maior eficiência computacional desta abordagem. Os autores também denotam que os resultados encontrados pela formulação com janela de tempo são considerados satisfatórios.

Qin, Du e Sha (2016) também abordaram o PAB com variação da maré ao longo do tempo, propondo um modelo de programação inteira (IP) e um modelo de programação por restrições (CP). A aplicação da CP no contexto do PAB é destacada como uma das principais contribuições do estudo, dada a escassez de trabalhos correlatos.

Os experimentos computacionais indicaram a superioridade da programação por restrições em comparação ao modelo IP em instâncias com pequenos domínios de soluções factíveis, bem como em instâncias de grande porte, que impedem a resolução do modelo por meio de *solver* computacional em tempo hábil. Os autores ainda pontuam que a incapacidade de provar a otimalidade das soluções obtidas pelo CP pode ser compensada pela utilização de um procedimento híbrido CP/IP.

Ursavas e Zhu (2016) apresentaram uma abordagem do PAB sob natureza estocástica, com incertezas inerentes aos tempos de chegada e atendimento. Sob estas considerações, o estudo visa encontrar as políticas ótimas de atendimento aos navios, minimizando os impactos relacionados aos cenários de incerteza. Para tal, os

autores desenvolveram um framework baseado em programação dinâmica estocástica.

Correcher e Alvarez-Valdes (2017) abordaram a integração do PAB com o QCAP considerando cais contínuo e chegadas dinâmicas. Os autores afirmam que uma das maiores limitações do problema tratado é o tamanho das instâncias que os métodos até então propostos são capazes de resolver. Diante desta limitação, os autores propuseram a aplicação de uma nova meta-heurística denominada *Biased Random-Key Genetic Algorithm* (BRKGA). O método proposto foi capaz de encontrar soluções ótimas em instâncias com até 40 navios, além de encontrar boas soluções em instâncias com até 100 navios, indicando a capacidade do algoritmo.

Ernst *et al.* (2017) propuseram dois novos modelos por programação linear inteira mista (MILP) para o PABC-D aplicado ao contexto de terminais de granéis sólidos. Ambas as formulações, que tem como objetivo minimizar o tempo total para completar o atendimento dos navios, também incluem considerações acerca da variação da maré, limitando a movimentação dos navios aos períodos de maré alta.

Os autores introduziram uma série de propriedades inerentes às soluções ótimas em conjunto com novas inequações, visando aumentar a eficiência computacional dos modelos. As instâncias de teste utilizadas foram geradas com base em dados coletados no terminal de granéis sólidos de Newcastle, na Austrália. Os modelos propostos foram capazes de encontrar soluções ótimas para 273 das 400 instâncias, com tempos de execução consideravelmente menores, demonstrando tanto a eficiência quanto a eficácia dos modelos.

Venturini *et al.* (2017) apresentaram uma extensão do tradicional PABD para um problema multi-portos, visando promover uma cooperação benéfica, tanto para os operadores dos navios quanto todos os portos incluídos na rede. A formulação por programação inteira (IP) apresenta tem como objetivos a minimização dos custos totais de ambas as partes, incluídos custos associados à ociosidade, a operação e aos atrasos. Tal modelo apresenta considerações acerca de velocidade de navegação dos navios, promovendo a análise do *tradeoff* entre minimização do tempo de total de operação e consumo de combustível.

Xiang, Liu e Miao (2017) apresentaram um novo modelo robusto bi-objetivo para o PABC-D, considerando incertezas associadas aos tempos de chegada e operação dos navios. O modelo proposto promove um *tradeoff* entre satisfação do cliente e minimização dos custos, sendo estes relacionados ao atraso para atracação e saída dos navios e os custos incorridos por atracar fora das posições ótimas definidas.

Para a resolução do problema, os autores desenvolveram uma meta-heurística denominada *Adaptive Grey Wolf Optimizer (AGWO)*, capaz de gerar um conjunto de soluções não dominadas, permitindo ao tomador de decisão escolher a programação adequada sob diferentes cenários.

Al-Hammadi e Diabat (2017) propuseram uma extensão ao trabalho de Robenek *et al.* (2014), apresentando uma nova formulação MILP para o PABH integrado ao YAP em portos multi-graneleiros com o objetivo de minimizar o atraso no atendimento aos navios. O novo modelo proposto contou com novas restrições, incluindo a definição de berços específicos para atender navios de alta tonelagem e a limitação de capacidade dos locais de armazenagem no pátio.

Lin, Ting e Wu (2018) apresentaram uma formulação para o PABC-D tendo como objetivos minimizar a soma ponderada dos tempos de atendimento e o distanciamento das posições ideais de atracação. Para a resolução do problema, os autores propuseram diferentes variantes do *Simulated Annealing (SA)*.

O SA foi utilizado em diversas instâncias de benchmarking e comparado com outros algoritmos da literatura e com o modelo executado em *solver* comercial. O método proposto foi capaz de superar diversos algoritmos, além de melhorar, na média, as soluções encontradas pelo *solver*. O algoritmo SA, segundo os autores, foi capaz de encontrar soluções de boa qualidade em todas as instâncias, atestando a sua validade.

Xiang, Liu e Miao (2018) abordaram novamente a natureza estocástica da operação portuária, desenvolvendo uma estratégia reativa para o problema integrado de alocação de berços discretos e alocação de guindastes. Esta abordagem parte de um planejamento inicial feito a partir de uma estratégia proativa, alterando-a ao longo do horizonte de planejamento conforme a variação dos tempos de chegada, operação e demais fatores vão ocorrendo, inviabilizando o plano inicial.

A formulação proposta da estratégia proativa tem como objetivos minimizar os custos associados à atracação fora da posição ótima, aos atrasos para atracação e término do atendimento ao navio e ao setup dos guindastes do cais. Para executar a estratégia reativa, os autores desenvolveram um método heurístico baseado na *Rolling Horizon Heuristic* (RHH). O método divide o horizonte de planejamento em sub-horizontes e resolve os subproblemas por meio de um método exato.

Agra e Oliveira (2018) propuseram novos modelos MIP para o PABC-D integrado à designação e programação dos guindastes do cais. Guindastes heterogêneos, com diferentes taxas de processamento, foram considerados nos modelos. Incluiu-se, também, a possibilidade de realocação de um guindaste de um navio em atendimento para outro, flexibilizando a programação. Os autores discutiram sobre aspectos relacionados a diferentes formulações e propuseram um modelo baseado na *Relative Position Formulation* (RPF), com o objetivo de minimizar o tempo total para a conclusão dos serviços.

Posteriormente, os autores promoveram uma discretização das variáveis relacionadas às dimensões de tempo e espaço, alterando o modelo. Esta formulação, denominada DRPF, permite a obtenção de melhores *lower bounds* na relaxação linear do problema. Uma terceira formulação, denominada DRPF+, foi proposta com base em novas modificações, obtendo os melhores valores de *lower bound* nos testes promovidos.

Para a resolução do problema em instâncias de grande porte, os autores desenvolveram uma heurística baseada na RHH, visando obter bons valores de *upper bound*. Tal abordagem permitiu a resolução de instâncias de tamanho prático a sua otimalidade em menos de uma hora, atestando a eficiência do método desenvolvido.

Wang *et al.* (2018) propuseram um modelo MILP para o PABD integrado com QCAP e YAP em terminais de transbordo de contêineres, visando minimizar os custos com o atraso ao atendimento aos navios e ao fluxo dos contêineres. Os autores desenvolveram uma heurística baseada em Geração de Colunas com técnicas de dupla estabilização para reduzir o tempo de execução do algoritmo. Os resultados obtidos tiveram um gap em torno de 1% em relação às soluções ótimas encontradas

pelo CPLEX, necessitando de um tempo significativamente menor para resolver o problema.

Ma *et al.* (2019) também trataram da integração do PAB com ambos os problemas (QCAP e YAP). Entretanto, o modelo MILP proposto considerou um tipo de layout para atracação no porto denominado como multi-contínuo. Segundo os autores, esta proposta permite uma melhor utilização da área de atracação em portos com descontinuidades do cais se comparada ao tratamento desta como layout híbrido ou discreto. Para a resolução de instâncias mais próximas a realidade dos terminais de contêineres, os autores desenvolveram uma meta-heurística *Guided Neighborhood Search* (GNS), obtendo boas soluções com tempos de execução significativamente baixos.

Unsal e Oguz (2019) estenderam o trabalho de Boland, Gulczynski e Savelsbergh (2012) propondo um modelo de programação linear inteira mista (MILP) e um algoritmo de decomposição de Benders para resolver o planejamento integrado de um terminal de exportação de carvão, considerando a alocação de berços de atracação, a designação de áreas de pátio para estocagem e a programação das empilhadeiras e recuperadoras. O método proposto decompõe o problema em problema mestre, formulado e resolvido como um modelo MILP, e subproblema, representado e resolvido como um modelo de programação por restrições (CP).

Mazioli *et al.* (2019) propuseram um modelo matemático para planejar a sequência de atracação de navios, baseado em cláusulas de contrato do regime *Charter Party*, especificamente o *Voyage Charter*. A formulação proposta pelos autores considera um porto com cais contínuo e um regime de chegadas dinâmicas, configurando um PABC-D.

No problema abordado, a empresa afretadora também é a proprietária do porto. Assim como em Ribeiro *et al.* (2016), foram considerados os pagamentos de *dispatch* e *demurrage* em função do *laytime*. Entretanto Mazioli *et al.* (2019) tiveram como foco a análise do impacto financeiro das cláusulas de *Charter Party*, em oposição ao próprio método de resolução. Ressalta-se também que, até então, não haviam pesquisas neste contexto aplicadas em formulações de cais contínuo.

O modelo matemático proposto, baseado em programação inteira, tem como objetivo maximizar o lucro, obtido em função dos recebimentos de *dispatch* e pagamentos de *demurrage*. Como método de solução, os autores propuseram uma meta-heurística baseada em SA.

Para a realização dos experimentos computacionais, os autores desenvolveram 300 instâncias baseadas nos dados obtidos no Terminal da Portocel, no Porto de Barra do Riacho. Nas 214 instâncias em que a execução do modelo em *solver* comercial encontrou a solução ótima, a meta-heurística desenvolvida conseguiu, em menor tempo computacional, os mesmos resultados.

Os autores analisaram a influência da negociação do contrato de *Voyage Charter* no resultado financeiro do porto, tomando como base cinco parâmetros: as definições de *layday*, *laytime*, *demurrage* e *dispatch* e, tempo padrão de operação. Discutiu-se também os impactos financeiros relacionados à manutenção do caís, além de analisar o a viabilidade de se investir na expansão do mesmo.

Conforme constatado pela revisão da literatura, há poucos artigos publicados que estudam o PAB em conjunto com YAP, principalmente com foco em terminais de carga a granel. A Tabela 1 apresenta algumas características dos artigos que tratam especificamente deste problema, permitindo identificar similaridades e diferenças entre eles e o trabalho desenvolvido por Bordal (2020), bem como a corrente pesquisa.

A coluna 1 identifica os trabalhos selecionados para a comparação. A coluna 2 apresenta o tipo de layout considerado para atracação no porto. A coluna 3 apresenta os tipos de decisões relativas à utilização dos pátios de armazenagem consideradas em cada artigo. A coluna 4 aponta os métodos empregados para a resolução dos problemas abordados. Por fim, as colunas 5 e 6 expõem o que foi considerado na função objetivo de cada um dos trabalhos.

Analisando a coluna 3, nota-se que os trabalhos de Robenek *et al.* (2014) e Al-Hammadi e Diabat (2017) se distanciam em relação ao problema de estudo tratado aqui. Ao contrário das demais pesquisas, os trabalhos citados não permitem a designação do momento e das posições em que cada pilha será formada nos pátios de estocagem. Os trabalhos destes, com foco em terminais multi-graneleiros,

estabelece que cada um dos pátios existentes será abastecido apenas com um determinado tipo de carga ao longo de todo o horizonte de planejamento tratado.

Tabela 1 – Classificação dos trabalhos similares à corrente pesquisa

Autores	Layout do Porto	Decisões de armazenagem	Método de Solução	FO	
				Tempo de permanência do navio no porto	Atraso na data de formação das pilhas
Boland <i>et al.</i> (2012)	Discreto	Momento e posição de formação de cada pilha nos pátios	Abordagem Híbrida (Heurística construtiva + IP)	X	
Robenek <i>et al.</i> (2014)	Híbrido	Tipo de granel armazenado em cada pátio	Branch and Price	X	
Babu <i>et al.</i> (2015)	Discreto	Momento e posição de formação de cada pilha nos pátios	Algoritmo construtivo guloso	X	
Al-Hammadi e Diabat (2017)	Híbrido	Tipo de granel armazenado em cada pátio	MILP	X	
Unsal e Oguz (2019)	Discreto	Momento e posição de formação de cada pilha nos pátios	Decomposição de Benders	X	
Bordal (2020)	Discreto / Contínuo	Momento e posição de formação de cada pilha nos pátios	MILP	X	X
Esta pesquisa	Discreto / Contínuo	Momento e posição de formação de cada pilha nos pátios	Simulated Annealing	X	X

Fonte: Próprio autor

Em relação aos artigos de Boland *et al.* (2012), Babu *et al.* (2015) e Unsal e Oguz (2019), o trabalho desenvolvido por Bordal (2020) e a corrente dissertação apresentam uma diferença substancial por considerar o plano de produção da usina de pelotização responsável pelo porto. Levando em conta as datas planejadas para formação dos lotes de pelota nos pátios, visando a manutenção do ritmo de produção da usina, a minimização da diferença entre a data planejada e a data efetiva de formação de cada pilha é incluída como um dos objetivos da FO no modelo de Bordal (2020) e na meta-heurística aqui proposta.

### 2.3 MODELO MATEMÁTICO PROPOSTO POR BORDAL (2020)

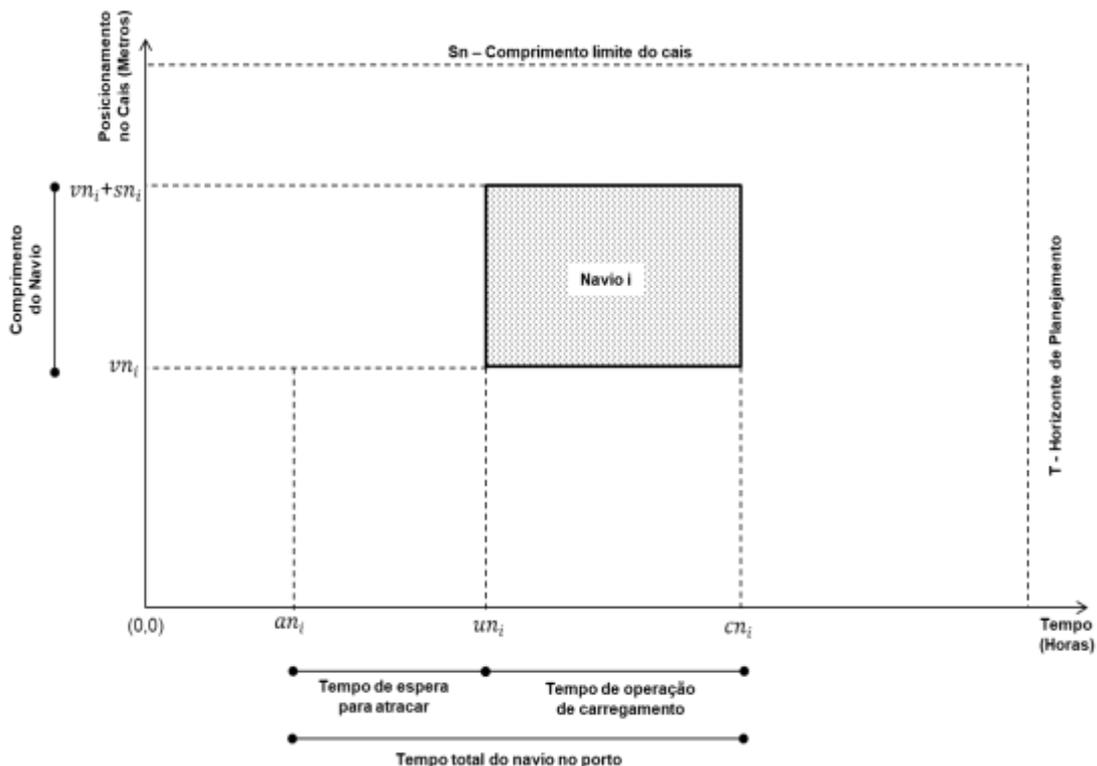
Neste subcapítulo é apresentado o modelo matemático proposto por Bordal (2020), que servirá de base para a meta-heurística a ser desenvolvida ao longo da corrente pesquisa. No que tange aos atributos temporais, o modelo adota um regime de chegadas dinâmicas, assumindo o ETA como o momento de chegada dos navios ao

porto. Em relação aos atributos espaciais, o modelo foi formulado de modo a permitir a utilização de instâncias tanto de berços discretos quanto de cais contínuo.

O modelo tem por objetivo planejar a alocação das áreas do pátio para ocupação da produção prevista e a melhor sequência e posicionamento no cais para a atracação dos navios. A função objetivo, apresentada na Equação 1, visa minimizar a soma entre a Parte I, que é o tempo total de permanência de todos os navios no porto, definido pela subtração do momento de desatracação do navio após seu carregamento menos o momento de chegada do navio (ETA), e a Parte II, que é o tempo total entre as datas planejadas para formação das pilhas de pelota nos pátios e os efetivos momentos em que as pilhas iniciaram a sua formação. Para permitir que seja dada maior ênfase no aspecto de produção das pilhas, o parâmetro  $\Psi$  é utilizado, multiplicando a Parte II da FO pelo peso determinada a ele.

A Figura 2 apresenta os principais parâmetros e variáveis de decisão em um diagrama espaço-tempo para a parte do modelo sobre o BAP (Parte I).

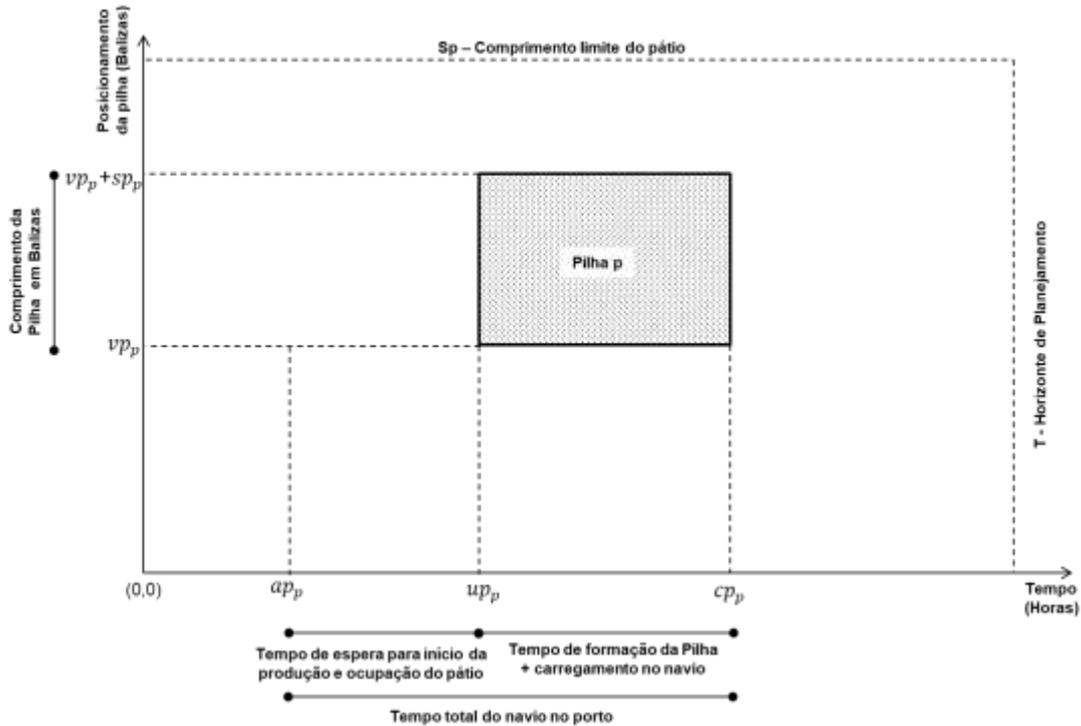
Figura 2 - Principais parâmetros e variáveis de decisão do modelo matemático da parte sobre o PAB



Fonte: Bordal (2020)

A Figura 3 apresenta os principais parâmetros e variáveis de decisão em um diagrama espaço-tempo para a parte do modelo sobre o Pátio (Parte II).

Figura 3 - Principais parâmetros e variáveis de decisão do modelo matemático da parte sobre o Pátio



Fonte: Bordal (2020)

O modelo matemático é apresentado a seguir em cinco partes: conjuntos, parâmetros, variáveis de decisão, função objetivo e restrições.

### Conjuntos

$NN$  - Conjunto de navios -  $NN = \{1, \dots, nn\}$ ;

$NP$  - Conjunto de pilhas -  $NP = \{1, \dots, np\}$ ;

$LN$  - Conjunto de seções do cais,  $LN = \{1, \dots, lc\}$ ;

$LP$  - Conjunto de seções de pátios,  $LP = \{1, \dots, na\}$ .

## Parâmetros

$T$ , horizonte de tempo de planejamento;

$pn_i$ , tempo de operação (carregamento) do navio  $i \in NN$  em unidade de tempo (hora);

$sn_i$ , comprimento do navio  $i \in NN$  em unidade de cais (1 metro), incluindo a folga entre navios;

$an_i$ , momento de chegada do navio  $i \in NN$  ao porto;

$pin_c$ , limite inicial da seção do cais  $c \in LN$  em unidade de cais (1 metro);

$pfnc$ , limite final da seção do cais  $c \in LN$  em unidade de cais (1 metro);

$pip_a$ , limite inicial da seção do pátio  $a \in LP$  em unidade de pátio (1 baliza);

$pfpa$ , limite final da seção do pátio  $a \in LP$  em unidade de pátio (1 baliza);

$pp_p$ , tempo de processamento pilha (alocação da produção)  $p \in NP$  no pátio unidade tempo (hora);

$sp_p$ , tamanho da pilha  $p \in NP$  em unidade de pátio (1 baliza);

$ap_p$ , momento de necessidade de área do pátio para a pilha  $p \in NP$  (unidade hora);

$pqn_p$ , define o número do navio ao qual a pilha  $p \in NP$  pertence;

$\Psi$ , peso utilizado na função objetivo para dar maior ênfase na redução do tempo de atraso na alocação da produção.

## Variáveis de decisão

$un_{ic}$ , momento de atracação do navio  $i \in NN$  na seção do cais  $c \in LN$ ;

$cn_{ic}$ , momento de desatracação do navio  $i \in NN$  na seção do cais  $c \in LN$ ;

$vn_{ic}$ , posição de atracação do navio  $i \in NN$  na seção do cais  $c \in LN$ ;

$up_{pa}$ , momento de início da alocação da produção da pilha  $p \in NP$  na seção do pátio  $a \in LP$ ;

$cp_{pa}$ , momento de liberação do pátio após o carregamento da pilha  $p \in NP$  no navio a qual pertence, na seção do pátio  $a \in LP$ ;

$vp_{pa}$ , posição inicial da pilha  $p \in NP$  em unidade de pátio (1 baliza), no cais  $a \in LP$ ;

Considerando-se o diagrama espaço-tempo com o tempo como abscissa e o cais como ordenada, têm-se as seguintes variáveis de decisão binárias:

$oln_{ijc}$ , assume valor igual a 1 se o retângulo do próximo navio  $j \in NN$  a atracar estiver totalmente acima do retângulo referente ao navio  $i \in NN$  no diagrama espaço-tempo e não houver sobreposição estando ambas no cais  $c \in LN$ ; 0, caso contrário;

$otn_{ijc}$ , assume valor igual a 1 se o retângulo do próximo navio  $j \in NN$  a atracar estiver totalmente à direita do retângulo referente ao navio  $i \in NN$  no diagrama espaço-tempo e não houver sobreposição estando ambas no cais  $c \in LN$ ; e 0, caso contrário;

$olp_{pqa}$ , assume valor igual a 1 se o retângulo do próxima pilha  $q \in NP$  a atracar estiver totalmente acima do retângulo referente ao navio  $p \in NP$  no diagrama espaço-tempo e não houver sobreposição estando ambas no pátio  $a \in LP$ ; e 0, caso contrário;

$otp_{pqa}$ , assume valor igual a 1 se o retângulo do próximo navio  $q \in NP$  a atracar estiver totalmente à direita do retângulo referente ao navio  $p \in NP$  no diagrama espaço-tempo e não houver sobreposição estando ambas no pátio  $a \in LP$ ; e 0, caso contrário;

$tn_{ic}$ , assume valor 1 se o navio  $i \in NN$  estiver na seção de cais  $c \in LN$ ; 0 caso contrário;

$tp_{pa}$ , assume valor 1 se a pilha  $p \in NP$  estiver no pátio  $a \in LP$ ; 0 caso contrário.

Nas restrições apresentadas a seguir, o símbolo  $\rightarrow$  representa uma restrição lógica, não linear, que é resolvida pelo *solver* CPLEX 12.8 utilizado para resolver o modelo matemático proposto.

## Função Objetivo

Minimizar

$$\Psi \sum_{p \in NP} \sum_{a \in LP} (up_{pa} - ap_p \quad tp_{pa}) + \sum_{i \in NN} \sum_{c \in LN} (cn_{ic} - an_i \quad tn_{ic}) \quad (1)$$

Parte I Parte II

## Restrições

Referente aos navios e os cais

$$\begin{aligned} (tn_{ic} = 1 \wedge tn_{jc} = 1) \rightarrow \\ (un_{jc} - un_{ic} - pn_i - (otn_{ijc} - 1) T \geq 0) \end{aligned} \quad \forall i, j \in NN, c \in LN, | i \neq j \quad (2)$$

$$\begin{aligned} (tn_{ic} = 1 \wedge tn_{jc} = 1) \rightarrow \\ (vn_{jc} - vn_{ic} - sn_i - (oln_{ijc} - 1) pfn_c) \geq 0 \end{aligned} \quad \forall i, j \in NN, c \in LN, | i \neq j \quad (3)$$

$$oln_{ijc} + oln_{jic} + otn_{ijc} + otn_{jic} \geq 1 \quad \forall i, j \in NN, c \in LN, | i \neq j \quad (4)$$

$$otn_{ijc} + otn_{jic} \geq 1 \quad \forall i, j \in NN, c \in LN, | i \neq j \quad (5)$$

$$(tn_{ic} = 1) \rightarrow (pn_i + un_{ic} - 1 = cn_{ic}) \quad \forall i \in NN, c \in LN \quad (6)$$

$$(tn_{ic} = 1) \rightarrow (un_{ic} \geq an_i) \quad \forall i \in NN, c \in LN \quad (7)$$

$$(tn_{ic} = 1) \rightarrow (un_{ic} \leq (T - pn_i + 1)) \quad \forall i \in NN, c \in LN \quad (8)$$

$$cn_{ic} \leq T \quad \forall i \in NN, c \in LN \quad (9)$$

$$\begin{aligned} (tn_{ic} = 1 \wedge tn_{jc} = 1) \rightarrow \\ (un_{jc} - cn_{ic} - (otn_{ijc} - 1) T \geq 0) \end{aligned} \quad \forall i, j \in NN, c \in LN, | i \neq j \quad (10)$$

$$(tn_{ic} = 0 \vee tn_{jc} = 0) \rightarrow (oln_{ijc} = 1) \quad \forall i, j \in NN, c \in LN \quad (11)$$

$$(tn_{ic} = 1 \wedge tn_{jc} = 1) \rightarrow (vn_{jc} - vn_{ic} - sn_i - (otn_{jic})(pfnc_c) \geq 0) \quad \forall i, j \in NN, c \in LN \quad (12)$$

$$(tn_{ic} = 1) \rightarrow (vn_{ic} \leq pfnc_c - sn_i) \quad \forall i \in NN, c \in LN \quad (13)$$

$$(tn_{ic} = 1) \rightarrow (vn_{ic} \geq pin_c) \quad \forall i \in NN, c \in LN \quad (14)$$

$$\sum_{c \in N} tn_{ic} = 1 \quad \forall i \in NN \quad (15)$$

$$(tn_{ic} = 0) \rightarrow (un_{ic} = 0) \quad \forall i \in NN, c \in LN \quad (15 a)$$

$$(tn_{ic} = 0) \rightarrow (cn_{ic} = 0) \quad \forall i \in NN, c \in LN \quad (15 b)$$

$$(tn_{ic} = 0) \rightarrow (vn_{ic} = 0) \quad \forall i \in NN, c \in LN \quad (15 c)$$

*Referente às pilhas e aos pátios*

$$(tp_{pa} = 1 \wedge tp_{qa} = 1) \rightarrow (up_{qa} - cp_{pa} - (otppqa - 1)T \geq 0) \quad \forall p, q \in NP, a \in LP, | p \neq q \quad (16)$$

$$(tp_{pa} = 1 \wedge tp_{qa} = 1) \rightarrow (vp_{qa} - vp_{pa} - sp_p - (olppqa - 1) pfp_a \geq 0) \quad \forall p, q \in NP, a \in LP, | p \neq q \quad (17)$$

$$(tp_{pa} = 0 \vee tn_{qa} = 0) \rightarrow (olppqa = 1) \quad \forall p, q \in NP, a \in LP \quad (18)$$

$$olppqa + olppqa + otppqa + otppqa \geq 1 \quad \forall p, q \in NP, a \in LP, | p \neq q \quad (19)$$

$$(tp_{pa} = 1) \rightarrow (up_{pa} \geq ap_p) \quad \forall p \in NP, a \in LP \quad (20)$$

$$(tp_{pa} = 1) \rightarrow (up_{pa} \leq (T - pp_p + 1)) \quad \forall p \in NP, a \in LP \quad (21)$$

$$(tp_{pa} = 1) \rightarrow (vp_{pa} \leq pfp_a - sp_p) \quad \forall p \in NP, a \in LP \quad (22)$$

$$(tp_{pa} = 1) \rightarrow (vp_{pa} \geq pip_a) \quad \forall p \in NP, a \in LP \quad (23)$$

$$(vp_{pa} \geq pfp_a + 1) \rightarrow (tp_{pa} = 0) \quad \forall p \in NP, a \in LP \quad (24)$$

$$\sum_{a \in LP} tp_{pa} = 1 \quad \forall p \in NP \quad (25)$$

$$(tp_{pa} = 1) \rightarrow (cp_{pa} \geq up_{pa} + pp_p - 1) \quad \forall p \in NP, a \in LP \quad (26)$$

$$cp_{pa} \leq T \quad \forall p \in NP, a \in LP \quad (27)$$

$$(vp_{pa} \geq pfp_a + 1) \rightarrow (tp_{pa} = 0) \quad \forall p \in NP, a \in LP \quad (28)$$

$$(vp_{pa} \geq pip_a \wedge vp_{pa} \leq pfp_a) \rightarrow (tp_{pa} = 1) \quad \forall p \in NP, a \in LP \quad (29)$$

$$(vp_{pa} \leq pip_a - 1) \rightarrow (tp_{pa} = 0) \quad \forall p \in NP, a \in LP \quad (30)$$

$$(tp_{pa} = 0) \rightarrow (up_{pa} = 0) \quad \forall p \in NP, a \in LP \quad (30 a)$$

$$(tp_{pa} = 0) \rightarrow (cp_{pa} = 0) \quad \forall p \in NP, a \in LP \quad (30 b)$$

$$(tp_{pa} = 0) \rightarrow (vp_{pa} = 0) \quad \forall p \in NP, a \in LP \quad (30 c)$$

*Referente à integração navio/cais e pilha/pátio*

$$(tn_{ic} = 1 \wedge tp_{pa} = 1) \rightarrow un_{ic} \geq up_{pa} + pp_p - 1 \quad \forall i \in NN, p \in NP, cn_{ic} \quad (31)$$

$$|pqn_p = i$$

$$(tn_{ic} = 1 \wedge tp_{pa} = 1) \rightarrow cp_{pa} = cn_{ic} \quad \forall i \in NN, p \in NP, cn_{ic}, a \in LP \quad (32)$$

$$|pqn_p = i$$

*Domínio das variáveis de decisão*

$$oln_{ijc}, otn_{ijc} \in \{0,1\} \quad i, j \in NN, c \in LN \quad (33)$$

$$olp_{pqa}, otn_{pqa} \in \{0,1\} \quad p, q \in NP, a \in LP \quad (34)$$

$$un_{ic}, vn_{ic}, cn_{ic} \in \mathbb{R}^+ \quad i \in NN, c \in LN \quad (35)$$

$$up_{pa}, vp_{pa}, cp_{pa} \in \mathbb{R}^+ \quad p \in NP, a \in LP \quad (36)$$

$$tn_{ic} \in \{0,1\} \quad i \in NN, c \in LN \quad (37)$$

$$tp_{pa} \in \{0,1\} \quad p \in NP, a \in LP \quad (38)$$

A Função Objetivo (FO), Equação (1), representa o tempo entre o momento de necessidade de área do pátio para alocação da produção para o momento em si de início da alocação da produção da pilha e sua disposição no pátio. Esta parte é multiplicada pelo parâmetro  $\Psi$  para dar maior ênfase a esta parte da FO. A segunda parte representa o tempo total dos navios no porto. Toda a FO deve ser minimizada.

Apresenta-se a seguir as restrições referentes aos navios/cais. As Restrições (2) garantem que no período não haja sobreposição na dimensão tempo dos navios no diagrama espaço-tempo.

As Restrições (3) garantem que caso dois navios estejam na mesma seção de cais, não haja sobreposição na dimensão espaço no mesmo diagrama, considerando a dimensão de cada pátio.

As Restrições (4) obrigam que pelo menos uma dessas variáveis seja igual a 1, garantindo assim, a atracação dos navios sem sobreposição com outros navios em tempo e espaço.

As Restrições (5) garantem que não haja sobreposição na dimensão tempo quando os navios estão na mesma seção de cais.

As Restrições (6) garantem que o tempo de termino do navio seja igual ao tempo de sua atracação somado ao tempo de sua operação de carregamento.

As Restrições (7) garantem que a atracação ocorra depois do momento de chegada do navio.

As Restrições (8) garantem que a atracação ocorra antes do limite de horizonte de tempo de planejamento menos o tempo de operação de carregamento do navio.

As Restrições (9) garantem que a desatracação do navio ocorra antes do término do horizonte de planejamento.

As Restrições (10) asseguram que caso dois navios estejam na mesma seção de cais, então, a atracação do navio posterior ocorre após a atracação do navio anterior.

As Restrições (11) garantem que se dois navios não estão na mesma seção de cais, então não haverá sobreposição entre eles.

As Restrições (12) asseguram que se dois navios estão na mesma seção de cais, avalia se a posição de atracação dos dois navios associadas a seus comprimentos é compatível com o limite final da seção do cais.

As Restrições (13) asseguram que se um navio está em uma seção de cais, então sua posição de atracação deve ser menor que o comprimento da seção menos seu próprio comprimento.

As Restrições (14) verificam, caso o navio esteja em uma seção do cais, que sua posição de atracação deve ser maior que o início da seção do cais.

As Restrições (15) garantem que um navio seja atracado em apenas em uma seção de cais.

As Restrições (15 a) asseguram que se um navio não está em uma seção de cais, então seu momento de atracação neste cais é zero.

As Restrições (15 b) asseguram que se um navio não está em uma seção de cais, então seu momento de desatracação do navio neste cais é zero.

As Restrições (15 c) asseguram que se um navio não está em uma seção de cais, então sua posição de atracação do navio neste cais é zero.

Apresenta-se a seguir as restrições referentes às pilhas/pátio.

As Restrições (16), caso duas pilhas estejam na mesma seção de pátio, garantem, então, que o início de armazenagem em tempo de uma deverá maior que o fim da outra.

As Restrições (17), caso duas pilhas estejam na mesma seção de pátio, garantem, então, que o início da localização de uma deverá maior que o fim da outra menos o seu comprimento

As Restrições (18) garantem que se pelo uma das duas pilhas não estão na mesma seção de pátio, então não haverá sobreposição entre elas.

As Restrições (19) obrigam que pelo menos uma dessas variáveis seja igual a 1, garantindo assim, que os lotes de pelotas ocupem balizas no pátio sem que haja sobreposição com outras balizas já utilizadas em tempo e espaço.

As Restrições (20) asseguram que a data de início do empilhamento no pátio seja maior ou igual à data de necessidade de alocação da produção desta pilha.

As Restrições (21) asseguram que a data de início do empilhamento no pátio seja menor ou igual que o horizonte de tempo de planejamento menos o tempo de armazenamento dela.

As Restrições (22) garantem que se uma pilha está numa seção do pátio, então sua posição final na seção deve ser menor que o fim da seção menos seu comprimento.

As Restrições (23) garantem que se uma pilha está numa seção do pátio, então sua posição inicial na seção deve ser maior que o início da seção.

As Restrições (24) garantem que se a posição inicial de uma pilha seja maior que a posição final de uma seção, então esta pilha não pode estar nesta seção.

As Restrições (25) asseguram que uma pilha pode estar em somente uma seção de pátio.

As Restrições (26) calculam que o tempo de fim de armazenagem da pilha seja igual ao seu tempo de início mais o tempo de armazenagem da pilha.

As Restrições (27) garantem que o fim de armazenagem seja menor que o horizonte de planejamento.

As Restrições (28) garantem que se a posição de atracação do navio está depois da posição final da seção do cais, o navio não está nesta seção do cais.

As Restrições (29) garantem que se a posição de atracação do navio está depois da posição inicial da seção do cais e antes da posição final do cais, o navio está nesta seção do cais.

As Restrições (30) garantem que se a posição de atracação do navio está antes da posição inicial da seção do cais, o navio não está nesta seção do cais.

As Restrições (30 a) asseguram que se uma pilha não está em uma seção de pátio, então seu momento de início da alocação da produção da pilha neste pátio é zero.

As Restrições (30 b) asseguram que se uma pilha não está em uma seção de pátio, então seu momento de liberação do pátio após o carregamento da pilha neste pátio é zero.

As Restrições (30 c) asseguram que se uma pilha não está em uma seção de pátio, então sua posição inicial da pilha neste pátio é zero.

Apresenta-se a seguir as restrições referentes à integração navio/cais e pilha/pátio.

As Restrições (31) definem que a atracação do navio só deve ocorrer após o início da formação da pilha acrescida ao seu tempo total de produção.

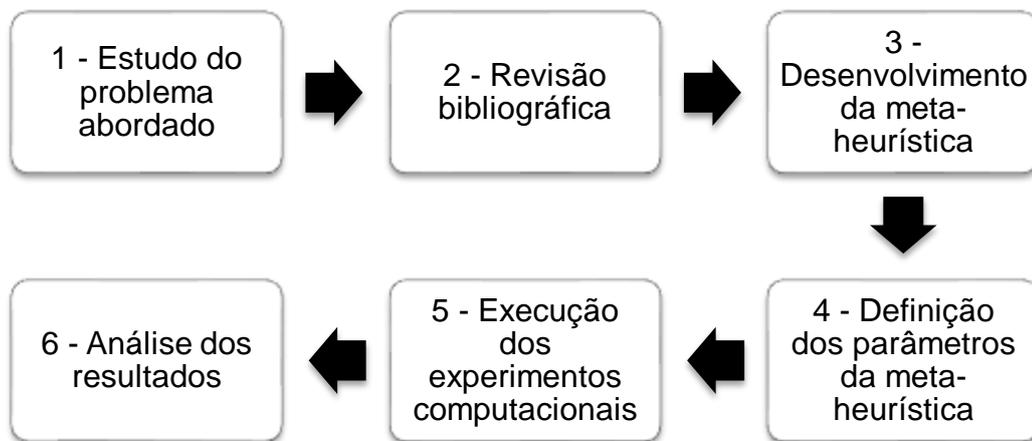
As Restrições (32) definem que as áreas alocadas pelas pilhas só sejam consideradas livres para um novo empilhamento após a desatracação do navio referido àquela pilha.

As Restrições (33) a (38) definem o domínio das variáveis de decisão.

### 3 MÉTODO DE PESQUISA

Neste capítulo serão apresentadas as etapas realizadas no desenvolvimento da corrente pesquisa. A execução do trabalho foi composta por seis etapas, conforme ilustrado esquematicamente na Figura 4. A seguir, cada uma destas etapas será apresentada.

Figura 4 - Etapas do projeto



Fonte: Próprio autor.

Conforme já indicado na introdução, o problema abordado (1) é o Porto de Ubu, que é pertencente à empresa Samarco. A descrição deste problema, apresentada no Capítulo 4, foi escrita com base no trabalho de Bordal (2020), que o abordou previamente, propondo um modelo matemático para a sua resolução.

A revisão bibliográfica (2) é constituída por pesquisas acerca dos temas relevantes ao trabalho. Para a realização do levantamento deste conteúdo, foram feitas pesquisas nas seguintes bases: Science Direct, Scopus, Web of Science (Coleção Principal) e Google Scholar. Utilizaram-se, a princípio, as palavras-chave “*Berth Allocation*”, “*Berth Scheduling*” e “*Berth Assignment*”, visando à obtenção de trabalhos que tenham como foco o problema de alocação de berços (PAB).

Posteriormente, as palavras-chave anteriores foram pesquisadas em conjunto com termos como “*Iron Ore*”, “*Bulk Cargo*” e “*Bulk Terminal*”, focando exclusivamente em

artigos que considerem o PAB em contextos semelhantes ao objeto de estudo da corrente pesquisa.

Por fim, a busca foi complementada com as palavras-chave “*Yard Assignment*”, “*Yard Allocation*” e “*Yard Management*”, para encontrar os artigos que tratem da designação de áreas dos pátios de armazenagem em conjunto com PAB. A revisão do portfólio selecionado é apresentada no capítulo 2 deste documento.

O desenvolvimento da meta-heurística (3) foi realizado em ambiente computacional Windows, utilizando a linguagem de programação C implementada no IDE Dev C++ 5.11 (BLOODSHED, 2019). Tomou-se como base para desenvolvimento da meta-heurística proposta neste trabalho, o *Simulated Annealing* (SA).

Para a meta-heurística proposta, os seguintes parâmetros, etapa (4), foram definidos: temperatura inicial ( $T$ ), coeficiente de resfriamento ( $\alpha$ ), número de iterações para cada temperatura ( $It$ ), temperatura de congelamento ( $fT$ ), número de reaquecimentos ( $Nr$ ).

Para esta definição, foi realizado um processo de calibração utilizando instâncias escolhidas aleatoriamente para promoção dos testes. Para definir cada parâmetro, a meta-heurística foi executada nas instâncias selecionadas, variando os valores atribuídos ao parâmetro em calibração, enquanto os demais foram mantidos fixos. Com base nos resultados obtidos, foram determinados quais valores seriam atribuídos a cada um dos parâmetros, levando em consideração a qualidade das soluções encontradas neste procedimento.

Após a definição dos parâmetros de entrada adequados, experimentos computacionais (5) com a meta-heurística desenvolvida foram realizados, utilizando as instâncias desenvolvidas por Bordal (2020). Para cada uma das instâncias, a meta-heurística foi executada 10 vezes, permitindo avaliar a sua estabilidade e a variabilidade dos resultados obtidos.

Por fim, a análise dos resultados (6) foi feita com objetivo de verificar a eficiência da meta-heurística proposta. Os resultados obtidos nas instâncias de teste pela meta-heurística foram comparados aos resultados encontrados pela execução do modelo matemático proposto por Bordal (2020). Os principais parâmetros analisados foram os valores encontrados da função objetivo e os tempos de execução em cada uma

das instâncias. Análises complementares foram promovidas, visando identificar o aproveitamento dos movimentos da meta-heurística e a sua convergência ao longo do período de execução.

## 4 PROBLEMA ESTUDADO E INSTÂNCIAS UTILIZADAS

Este capítulo apresenta o problema tratado por Bordal (2020), referente ao Porto de Ubu, pertencente à empresa Samarco Mineração S/A. Ao final, é feita a descrição das instâncias utilizadas para a avaliação da meta-heurística desenvolvida nesta pesquisa.

### 4.1 PROBLEMA ESTUDADO

O processo produtivo da Samarco Mineração S/A contempla os seguintes processos, brevemente descritos ao longo deste subcapítulo: 1) extração e 2) beneficiamento do minério de ferro; 3) transporte de polpa de minério de ferro entre os estados de Minas Gerais e Espírito Santo; 4) pelotização e 5) embarque do produto final (SAMARCO, 2015).

As primeiras etapas do processo ocorrem na unidade de Germano, localizada entre os municípios de Mariana e Ouro Preto, em Minas Gerais. O minério de ferro é extraído de minas a céu aberto e conduzido, por meio de correias transportadoras, aos concentradores, onde são realizadas as operações de beneficiamento.

A etapa de beneficiamento visa à adequação do minério às especificações cabíveis para as próximas etapas do processo produtivo. Por meio das operações de britagem, moagem, deslamagem e flotação, obtêm-se um concentrado com maior teor de ferro. Acrescido de 30% de água, o minério, em estado de polpa, é transportado por meio de minerodutos com aproximadamente 400 quilômetros de extensão até a unidade de Ubu, no município de Anchieta (ES).

Na unidade de Ubu, a polpa de minério de ferro passa inicialmente por processos de espessamento e filtragem para retirada da água. Posteriormente, adicionam-se insumos para permitir o pelotamento do material. Para adquirir as propriedades desejáveis, as pelotas são endurecidas e finalizadas em tratamento térmico nos fornos da planta de processamento.

Por fim, o produto acabado é armazenado nos pátios de armazenagem até o momento do seu escoamento, realizado por meio do Porto de Ubu. A Figura 5 apresenta a vista aérea da planta de processamento em Anchieta (ES), destacando a localização dos pátios de armazenagem e o Porto de Ubu.

Figura 5 - Vista aérea da planta de processamento da Samarco em Anchieta-ES



Fonte: Bordal (2020).

A área de armazenagem das pelotas é composta por três pátios distintos (Pátio A, Pátio B e Pátio C), evidenciados na Figura 6, que somados possuem uma capacidade total de armazenagem estimada em 1.800.000 toneladas. Cada pátio possui subdivisões por balizas a cada 10 metros. De acordo com o seu volume, os lotes de pelota são alocados a uma área delimitada por um conjunto de balizas, sendo que a ocupação parcial de uma baliza é considerada como utilização inteira (BORDAL, 2020). O Porto de Ubu é composto por dois berços de capacidades distintas, alimentados por um sistema único de carregamento de navios. Atualmente apenas um berço discreto está em operação. Entretanto, nas instâncias utilizadas, são consideradas diversas configurações de cais, incluindo a utilização de dois berços discretos.

Figura 6 - Pátios de armazenagem de pelotas da Samarco

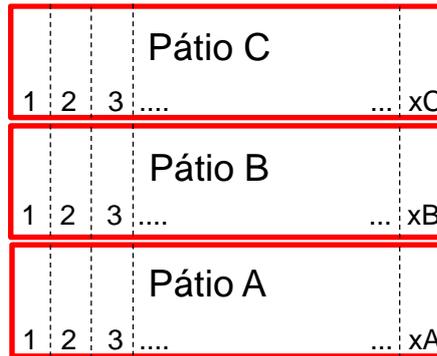


Fonte: Bordal (2020).

O modelo matemático proposto por Bordal (2020), apresentado no referencial teórico desta pesquisa, permite realizar a alocação dos lotes de pelota em pilhas no pátio de forma integrada ao sequenciamento das operações de embarque e carregamento dos navios no porto. A análise dos resultados do modelo permite avaliar o equilíbrio entre a produção dos lotes de pelotas e o escoamento desta produção no porto.

Apesar de a empresa estudada possuir três pátios distintos, representados graficamente na Figura 7, instâncias com um único pátio disponível e com capacidade parcialmente ocupada foram propostas por Bordal (2020) para poder abordar diversos cenários e analisar o impacto destas situações no planejamento da operação portuária.

Figura 7 - Representação gráfica das balizas do pátio real da Samarco



Fonte: Bordal (2020).

O modelo matemático também assume diversas configurações de cais para atracação dos navios, contrapondo a utilização de um único berço discreto, realidade atual do porto. Outra premissa desta formulação é a consideração do ETA de cada navio como a sua data de prontidão para atracação, denominada *Notice of Readiness* (NOR). Destaca-se também que cada navio só pode ser admitido para atracação após o término da produção e armazenagem de seu respectivo lote.

Na seção a seguir, são apresentadas as instâncias formuladas por Bordal (2020) para avaliação do seu modelo matemático, e que também serão utilizadas pela corrente pesquisa.

#### 4.2 INSTÂNCIAS UTILIZADAS

A utilização das instâncias propostas por Bordal (2020) permite a comparação da meta-heurística com os resultados obtidos pela resolução do modelo matemático por meio de um método exato, o que contribui para a validação da meta-heurística desenvolvida. A seguir, estas instâncias são apresentadas.

Foram consideradas 25 instâncias, que abordam duas premissas diferentes em termos de horizonte de tempo. As instâncias referentes ao horizonte de curto prazo, com planejamento estipulado entre 2 e 3 semanas, consideram um horizonte de tempo de 504 horas. As instâncias de médio prazo, que levam em consideração um planejamento de dois meses, limitam-se a um horizonte de tempo de 1440 horas.

Em conjunto com o limite do horizonte de tempo estabelecido, as instâncias de curto e médio prazo se diferenciam em termos de navios a serem atendidos. Todas as instâncias tratadas como de curto prazo consideram um total de 10 navios a serem atendidos. Para as instâncias de médio prazo, é definido o atendimento de 42 navios.

Além do horizonte de tempo tratado, outras características foram consideradas para elaboração das instâncias. Uma delas é a quantidade de pilhas para cada navio. Para uma mesma carga destinada a um navio, algumas instâncias consideram que as pelotas serão empilhadas em uma única pilha. Em outras instâncias, considera-se a divisão da carga para formação em duas pilhas. Ao considerar o carregamento em duas pilhas, a taxa de carregamento do navio é menor.

As instâncias também se diferenciam em termos de quantidade de pátios disponível para formação das pilhas. Algumas instâncias consideram apenas uma parcela disponível do pátio para testar o comportamento do modelo quando ocorre uma maior restrição de área para estocagem.

A quantidade de berços disponíveis para atracação dos navios também é variada na elaboração das instâncias. Apesar do porto estudado não permitir atualmente o carregamento de mais de um navio, a pesquisa considera a capacidade de carregar dois navios atracados ao mesmo tempo.

A Tabela 2 apresenta as 25 instâncias desenvolvidas por Bordal (2020) e suas características. As instâncias foram divididas em 7 grupos em função das diferenças entre suas características. A coluna (1) indica estes grupos de instâncias que serão testados. As colunas (2) e (3) indicam os números e os códigos que representam as instâncias.

As colunas de (4) a (8) apresentam as características de cada instância. A coluna (4) representa a quantidade de navios utilizados para cada instância. A coluna (5) representa a quantidade de pilhas a serem alocadas nos pátios de armazenagem de pelotas. A coluna (6) representa a quantidade de pátios e seu percentual de área disponível para armazenagem. A coluna (7) indica a quantidade de berços disponíveis para atracação. A coluna (8) representa o valor do parâmetro  $\Psi$  utilizado em cada instância.

Tabela 2 – Instâncias utilizadas

Grupo	Nº Instância	Código da Instância	Quant. Navio	Quant. Pilha	Quant. / Disponibilidade dos Pátios	Quant. Berços	Valor do $\Psi$
1	1	10N_10P_1PA-100%_1BD_ψ1	10	10	1 - 100%	1	1
	2	10N_10P_1PA-100%_2BD_ψ1	10	10	1 - 100%	2	1
	3	10N_10P_3PA-100%_1BD_ψ1	10	10	3 - 100%	1	1
	4	10N_10P_3PA-100%_2BD_ψ1	10	10	3 - 100%	2	1
2	5	10N_20P_1PA-100%_1BD_ψ1	10	20	1 - 100%	1	1
	6	10N_20P_1PA-100%_2BD_ψ1	10	20	1 - 100%	2	1
	7	10N_20P_3PA-100%_1BD_ψ1	10	20	3 - 100%	1	1
	8	10N_20P_3PA-100%_2BD_ψ1	10	20	3 - 100%	2	1
3	9	42N_42P_1PA-100%_1BD_ψ1	42	42	1 - 100%	1	1
	10	42N_42P_1PA-100%_2BD_ψ1	42	42	1 - 100%	2	1
	11	42N_42P_3PA-100%_1BD_ψ1	42	42	3 - 100%	1	1
	12	42N_42P_3PA-100%_2BD_ψ1	42	42	3 - 100%	2	1
4	13	42N_84P_1PA-100%_1BD_ψ1	42	84	1 - 100%	1	1
	14	42N_84P_1PA-100%_2BD_ψ1	42	84	1 - 100%	2	1
	15	42N_84P_3PA-100%_1BD_ψ1	42	84	3 - 100%	1	1
	16	42N_84P_3PA-100%_2BD_ψ1	42	84	3 - 100%	2	1
5	17	10N_10P-50%_1PA_1BD_ψ1	10	10	1 - 50%	1	1
	18	10N_10P_1PA-35%_1BD_ψ1	10	10	1 - 35%	1	1
	19	10N_10P_1PA-27%_1BD_ψ1	10	10	1 - 27%	1	1
6	20	10N_10P_1PA-50%_1BD_ψ10	10	10	1 - 50%	1	10
	21	10N_10P_1PA-35%_1BD_ψ10	10	10	1 - 35%	1	10
	22	10N_10P_1PA-27%_1BD_ψ10	10	10	1 - 27%	1	10
7	23	10N_10P_1PA-50%_1BD_ψ0	10	10	1 - 50%	1	0
	24	10N_10P_1PA-35%_1BD_ψ0	10	10	1 - 35%	1	0
	25	10N_10P_1PA-27%_1BD_ψ0	10	10	1 - 27%	1	0

Fonte: Bordal (2020).

A seguir, são descritos cada um dos 7 grupos. As instâncias do Grupo 1, que são definidas como de curto prazo, consideram 10 navios a serem carregados com pelotas armazenadas em pilhas únicas disponibilizadas nos pátios de armazenagem, totalizando 10 pilhas. Neste tipo de instância, a taxa de carregamento é menor em função da limitação de carregamento das pelotas por uma única recuperadora. As instâncias se diferenciam entre si pelo número de pátios e berços disponíveis.

O Grupo 2 replica as características do primeiro grupo, porém considera que as pelotas destinadas a cada um dos 10 navios sejam armazenadas em duas pilhas

distintas, totalizando 20 pilhas a serem empilhadas nos pátios. Esta configuração permite uma taxa maior de carregamento dos navios por permitir que a recuperação das pelotas seja feita por dois equipamentos operando simultaneamente.

O Grupo 3 é um grupo de instâncias baseadas no planejamento de longo prazo. Este grupo considera a existência de 42 navios a serem atendidos dentro do horizonte de tempo tratado. Cada navio é abastecido por pelotas armazenadas em pilhas únicas disponibilizadas nos pátios de armazenagem. Logo, há um total de 42 pilhas que deverão ser formadas nos pátios.

O Grupo 4 considera que os mesmos 42 navios serão abastecidos por pelotas armazenadas em duas pilhas distintas. Conseqüentemente, é considerada uma maior taxa de carregamento nos navios, tendo como base a utilização simultânea de dois equipamentos para abastecer o carregador de navios.

Todas as instâncias dos Grupos 1, 2, 3 e 4 consideram uma disponibilidade de 100% dos pátios para armazenagem. Além disso, o parâmetro  $\Psi$  sempre foi considerado como valor 1. Variando estes parâmetros, as demais instâncias que compõem os Grupos 5, 6 e 7 foram criadas para avaliar o efeito da menor disponibilidade de área nos pátios, e também o efeito da alteração da prioridade dada à redução do atraso no início de formação das pilhas, nas soluções finais obtidas.

No Grupo 5, as instâncias elaboradas consideram um único pátio disponível para armazenagem das pelotas, com percentuais diferentes de disponibilidade. Nas instâncias deste grupo, o parâmetro  $\Psi$  foi mantido com valor igual a um 1. As instâncias dos Grupos 6 e 7 são análogas às instâncias do Grupo 5, considerando redução de disponibilidade de área para armazenagem. Os Grupos se diferenciam em função do parâmetro  $\Psi$ . No Grupo 6, o parâmetro  $\Psi$  foi definido com valor igual a 10, aumentando a prioridade em reduzir o atraso de formação das pilhas e, conseqüentemente, reduzindo a importância do atraso da atracação dos navios no porto. No Grupo 7, o parâmetro  $\Psi$  foi colocado com valor igual a 0, anulando a importância do atraso de formação das pilhas no cálculo da função objetivo. Neste caso, o modelo matemático e a meta-heurística priorizarão o atendimento aos navios.

## 5 DESENVOLVIMENTO DA META-HEURÍSTICA

Após a apresentação do problema abordado e a fundamentação teórica cabível, será apresentada, nesta etapa, a meta-heurística proposta para resolver o planejamento da fila de navios integrado à designação de áreas de pátio de armazenagem para o atendimento do plano de produção da usina de pelotização. A meta-heurística considera as mesmas variáveis de decisão e o mesmo objetivo do modelo matemático de Bordal (2020), visando à minimização conjunta do tempo total de permanência dos navios no porto e o tempo total de disponibilização da área nos pátios para os lotes de pelota.

A meta-heurística proposta foi desenvolvida com base no *Simulated Annealing* (SA). Este método teve a sua utilização popularizada para a resolução de problemas de otimização combinatória por Kirkpatrick *et al.* (1983). O SA é baseado em um tratamento térmico utilizado na metalurgia conhecido como *annealing*, ou arrefecimento, um processo de cristalização dos metais por resfriamento.

No processo de *annealing*, o metal aquecido é resfriado lentamente, promovendo uma cristalização uniforme e com menor estado de energia, o que reduz os seus defeitos. Um menor tempo de resfriamento implica em uma cristalização prejudicada, e, conseqüentemente, um material defeituoso. A otimização por meio da meta-heurística *Simulated Annealing* segue uma ideia análoga à cristalização por resfriamento (LIN; TING; WU, 2018).

O SA parte de uma solução inicial, que pode ser gerada aleatoriamente, e realiza um procedimento iterativo de busca local, encontrando novas soluções a cada iteração. O valor obtido pela nova solução é comparado a solução atual, visando determinar se houve melhoria. Caso a nova solução seja superior, esta passa a ser nova solução vigente e o processo de busca continua utilizando-a como base.

Se a solução gerada durante o processo de busca for inferior à solução atual, ela ainda poderá ser aceita. Isto ocorre com o objetivo de impedir que o procedimento se prenda a um mínimo local e deixe de explorar novas soluções. A probabilidade de aceitação desta solução é dada a partir do critério proposto em Metropolis *et al.* (1953),

que varia em função da temperatura simulada do sistema. No começo do SA, onde este parâmetro se encontra mais elevado, as chances de se aceitar uma solução inferior são maiores. Ao longo das iterações, a temperatura decresce e, portanto, a probabilidade de aceite diminui. Ao final, a meta-heurística tende apenas a aprimorar a solução até que o critério de parada seja alcançado.

Nas próximas seções, serão detalhados os seguintes aspectos pertinentes para a compreensão da meta-heurística implementada: a representação dos pátios e do cais; os parâmetros, variáveis e estruturas utilizadas; a visão geral da meta-heurística proposta e; o detalhamento das principais funções desenvolvidas.

### 5.1 REPRESENTAÇÃO DOS PÁTIOS E DO CAIS

Para o desenvolvimento da meta-heurística, o cais do porto e os pátios de armazenagem serão representados por meio de diagramas espaço-tempo. A Figura 2 ilustra a representação referente ao cais. Este diagrama possui uma área retangular definida a partir da coordenada  $(0,0)$  até a coordenada  $(tH, qL)$ , onde  $tH$  é o horizonte de tempo considerado e  $qL$  é o comprimento total do cais.

Cada navio é representado no diagrama por um retângulo, cuja largura equivale ao seu tempo de operação estimado, e sua altura corresponde ao seu comprimento máximo, ou *Length Overall* (LOA). Assim, uma solução factível para o planejamento da fila de navios deverá conter todos os navios posicionados no diagrama. Além disso, para que o arranjo seja factível, um navio não poderá ser alocado no eixo X (tempo) em uma coordenada inferior ao seu ETA.

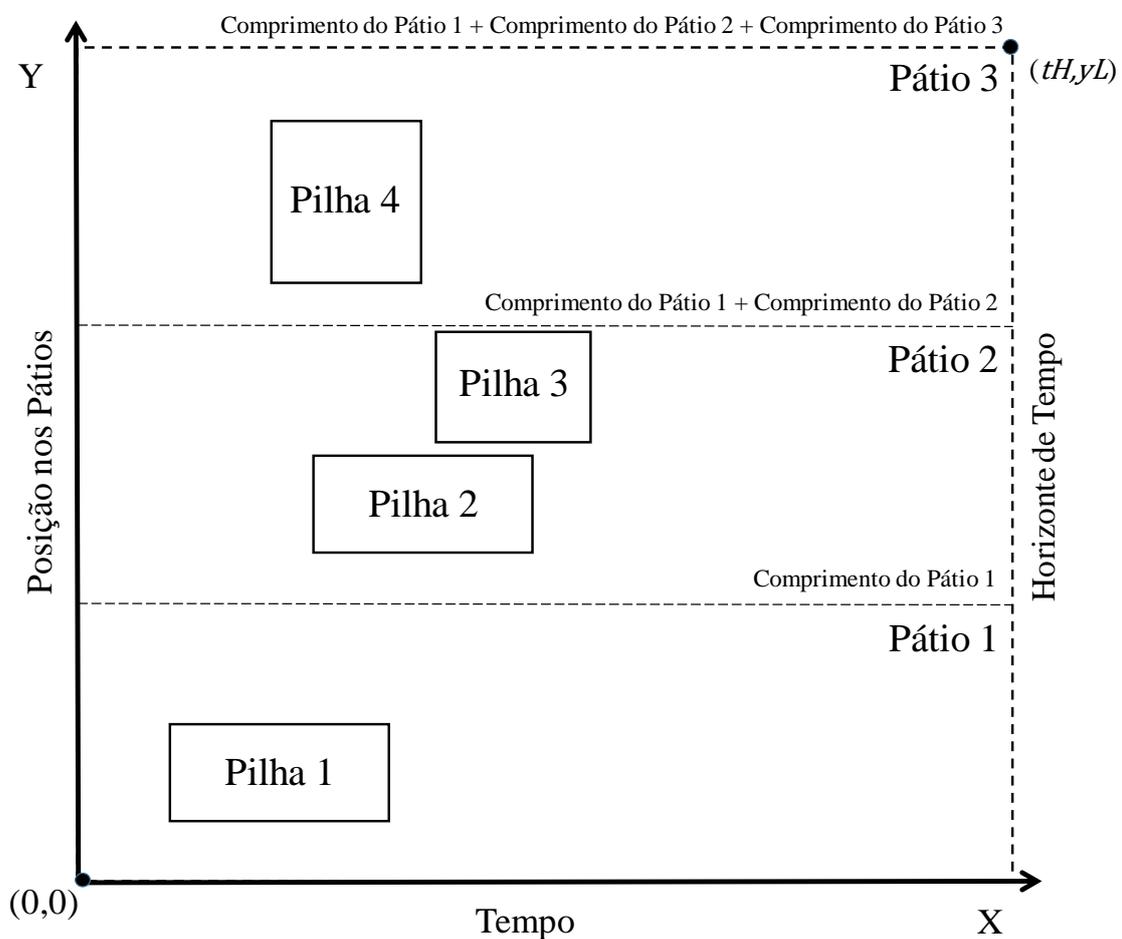
De maneira análoga à representação do cais do porto, a Figura 3 ilustra o diagrama espaço-tempo para o pátio de armazenagem. O diagrama possui uma área retangular definida a partir da coordenada  $(0,0)$  até a coordenada  $(tH, yL)$ , onde  $tH$  é o horizonte de tempo considerado e  $yL$  é o comprimento total do pátio.

Cada pilha também é representada no diagrama por um retângulo. O comprimento equivale à soma dos tempos necessários para formar a pilha no pátio e carregá-la no navio, enquanto a altura representa o comprimento total da pilha. Considera-se como

premissa que a largura de cada pilha é equivalente a largura total do pátio, tornando irrelevante a sua representação no diagrama espaço-tempo.

Dado que várias das instâncias tratadas retratam os três pátios da usina de pelotização, faz-se necessário adaptar o diagrama apresentado na Figura 3 para uma representação mais fidedigna, considerando cada um dos pátios separadamente. A Figura 8 ilustra esta nova representação, dividindo o diagrama em três retângulos, cujas alturas equivalem aos comprimentos de cada pátio.

Figura 8 - Representação gráfica das balizas do pátio real da Samarco



Fonte: Próprio autor.

Considera-se como premissa que cada pilha pode ser alocada somente em um dos pátios. Assim, um arranjo factível deverá conter todas as pilhas encaixadas de modo que não sobreponham os limites dos pátios. Ressalta-se também que, analogamente à relação entre as posições dos navios e seus respectivos ETAs, uma pilha não pode ser posicionada antes da data de necessidade de área do pátio.

Ao tratar do planejamento da fila de navios integrado com a designação de áreas dos pátios, a seguinte premissa é adotada nesta pesquisa: um navio só poderá atracar no cais quando todas as pilhas correspondentes a ele já estiverem formadas no pátio. Assim, para que uma solução seja factível, todos os navios deverão ser posicionados no diagrama do cais em um instante igual ou superior ao momento de formação suas respectivas pilhas no diagrama dos pátios.

## 5.2 PARÂMETROS, ESTRUTURAS E VARIÁVEIS DESENVOLVIDAS

Nesse subcapítulo, são definidos os principais parâmetros, estruturas e variáveis da meta-heurística proposta. Posteriormente, as estruturas e suas variáveis são detalhadas.

Os parâmetros de entrada necessários para o funcionamento da meta-heurística são apresentados a seguir.

$T_0$	Temperatura inicial
$\alpha$	Coefficiente de resfriamento
$It$	Número de iterações para cada temperatura
$fT$	Temperatura de congelamento
$Nr$	Número de reaquecimentos

Para definir o tamanho dos principais vetores e matrizes da meta-heurística, são utilizados os seguintes parâmetros de entrada:

$h\_tempo$	Horizonte máximo de tempo
$num\_navios$	Número de navios considerados na instância
$num\_pilhas$	Número de pilhas consideradas na instância
$num\_bercos$	Número de berços disponíveis
$num\_patios$	Número de pátios disponíveis

*disp\_berco* [*nb*] Comprimento disponível em cada um dos berços

*disp\_patio* [*np*] Comprimento disponível para armazenagem em cada um dos pátios

Além dos parâmetros citados, ainda foi incluído o *max\_pilhas\_navio*, que corresponde ao número máximo de pilhas destinadas a um mesmo navio. Este parâmetro é utilizado na estrutura *navio* e nas funções responsáveis pelo posicionamento das pilhas nos pátios e dos navios nos berços, respectivamente, que serão apresentadas no decorrer deste capítulo.

A seguir serão apresentadas as estruturas e todas as variáveis que as compõem:

A estrutura *Ship* (Figura 9) representa os navios, sendo composta por seis parâmetros. O parâmetro *id\_n* serve como identificador de cada um dos navios. Os parâmetros *operacao*, *comprimento* e *ETA* representam, respectivamente, o tempo necessário para carregar um navio após a sua atracação, o seu comprimento, e a sua data estimada de chegada, adotada como a chegada efetiva na resolução do problema. O *pilhas\_pertencentes*[*mpn*] é um vetor que contém as identificações de cada uma das pilhas que deverão ser carregadas nos respectivos navios. O parâmetro *min\_atracacao* assume o momento mais cedo em que o navio pode atracar, levando em consideração o ETA do navio e o menor tempo possível para formação de suas pilhas. Este parâmetro é utilizado na função *OrderShip* responsável pela ordenação dos navios que precede o processo de geração da solução inicial do problema.

Figura 9 – Estrutura *Ship*

```
Ship{
    int id_n;
    int operacao;
    int comprimento;
    int ETA;
    int pilhas_pertencentes[mpn]; // mpn varia de 1 a max_pilhas_navio;
    int min_atracacao;
};
```

Fonte: Próprio autor.

A estrutura *Pile* (Figura 10) representa cada uma das pilhas e é composta por seis parâmetros. O parâmetro *id\_p* é responsável pela identificação de cada pilha. Os parâmetros *formacao*, *tamanho* e *data\_necessidade* representam, respectivamente, o tempo necessário para formar a pilha, o comprimento ocupado no pátio por ela, e o instante em que sua formação pode ser iniciada, considerando o plano de produção. Os parâmetros *navio\_pertence* e *navio\_carregamento* designam, respectivamente o navio no qual a pilha em questão deverá ser carregada e o tempo estipulado para o carregamento deste navio. Considerando a premissa de que uma área de pátio ocupada por uma pilha só é liberada após o carregamento desta, este último parâmetro é um dos fatores que compõem o tempo de permanência de uma pilha no pátio.

Figura 10 – Estrutura *Pile*.

```
Pile{
    int id_p;
    int formacao;
    int tamanho;
    int data_necessidade;
    int navio_pertencente;
    int navio_carregamento;
};
```

Fonte: Próprio autor.

A estrutura *Solution* (Figura 11) é utilizada para representar alguns aspectos de uma solução gerada pelas rodadas de execução da meta-heurística. No decorrer da meta-heurística, três tipos de solução são considerados: melhor solução (*Sb*), solução atual (*Sc*) e solução vizinha (*Sn*).

Figura 11 – Estrutura *Solution*

```
Solution{
    int FO;
    int vet_pilhas[n_pil]; // n_pil varia de 1 a num_pilhas;
    int vet_navios[n_nav]; // n_nav varia de 1 a num_navios;
};
```

Fonte: Próprio autor.

A variável *FO* serve para armazenar o valor da função objetivo da solução encontrada. O vetor *vet\_pilhas[n\_pil]* é utilizado para armazenar a ordem pela qual as pilhas serão posicionadas no pátio pelas funções de criação do arranjo de pilhas. O vetor *vet\_navios[n\_nav]* é utilizado para armazenar a ordem pela qual os navios serão posicionados no cais pelas funções de arranjo dos navios.

Para representar as soluções geradas ao longo da execução da meta-heurística, duas estruturas, *Sol\_Pile* e *Sol\_Ship*, são utilizadas para capturar as coordenadas das pilhas e dos navios, respectivamente, nos seus diagramas espaço-tempo. Na estrutura *Sol\_Pile* (Figura 12), as variáveis *patio*, *x* e *y* representam, respectivamente, o pátio em que a pilha se encontra e sua posição em relação aos eixos tempo e comprimento do diagrama neste pátio.

Figura 12 – Estrutura *Sol\_Pile*.

```
Sol_Pile{
    int patio;
    int x;
    int y;
    int direcao;
    int t_espera;
};
```

Fonte: Próprio autor.

A variável *direcao*, que pode assumir os valores 0 ou 1, influi no modo como é feito o arranjo das pilhas no diagrama dos pátios. Normalmente, o processo de arranjo começa a sua busca ao longo do eixo Y a partir da coordenada zero e vai subindo ao longo do diagrama do pátio. Quando a variável *direcao* assumir valor 0, as pilhas em questão serão selecionadas seguindo esta lógica. Ao modificar o valor da *direcao* para 1, a meta-heurística inverterá a direção em relação ao eixo Y pela qual a pilha será posicionada, iniciando a partir do final do pátio e descendo até a coordenada inicial. A variável *t\_espera* captura a diferença entre o instante em que uma pilha termina de ser formada no pátio e o instante em que se inicia o carregamento do seu respectivo navio.

Na estrutura *Sol\_Ship* (Figura 13), as variáveis *berco*, *x* e *y* representam, respectivamente, o berço em que o navio irá atracar e sua posição em relação aos eixos tempo e comprimento do berço.

Figura 13 – Estrutura *Sol\_Ship*

```
Sol_Ship{
    int berco;
    int x;
    int y;
};
```

Fonte: Próprio autor.

### 5.3 VISÃO GERAL DA META-HEURÍSTICA

Neste subcapítulo, é apresentada a visão geral da meta-heurística, cujo pseudocódigo é exibido na Figura 14. No início da meta-heurística, a função *InitializeParameters()* inicializa os parâmetros e variáveis necessários para a sua execução. Os dados específicos de cada instância, como o horizonte de tempo e as características dos pátios e do cais, são introduzidos por meio da função *ReadData()*. Os vetores de estruturas *navios[n\_nav]* e *pilhas[n\_pil]*, definidos, respectivamente, como tipos *Ship* e *Pile*, são gerados dentro desta função, organizando os dados referentes aos navios e às pilhas.

A próxima etapa consiste na construção da solução inicial. Primeiramente, a função *OrderShip()* é chamada para promover a ordenação dos navios, em ordem crescente, com base no parâmetro *min.atracacao*. Assim, os navios que podem atracar o mais cedo possível em relação aos demais, considerando as restrições inerentes ao problema, terão prioridade na função responsável pelo arranjo dos navios no diagrama do cais da solução inicial, sendo os primeiros a atracar.

Posteriormente, as pilhas também são ordenadas por meio da função *OrderPile()*. Considerando a premissa de que um navio só poderá atracar no cais após o instante em que suas respectivas pilhas forem formadas nos pátios, a ordenação das pilhas por meio desta função é feita com base na ordem dos navios gerada pela função

*OrderShip()*. As pilhas referentes aos navios com menor valor no parâmetro *min.atracacao* serão as primeiras da lista utilizada na criação da solução inicial. Em instâncias com mais de uma pilha prevista por navio, o valor do parâmetro *data\_necessidade* determinará qual das pilhas de um navio será listada primeiro.

Figura 14 - Pseudocódigo da meta-heurística proposta

```

1 InitializeParameters()
2 ReadData()
3 OrderShip()
4 OrderPile()
5 Sc = CreateSolution()
6 Sc.OF = CalculateOF(Sc)
7 CopySolution(Sb,Sc)
8 T0 = T
9 r = 1
10 WHILE (r <= Nr)
11     Iter = 1
12     WHILE (T > fT)
13         WHILE (Iter <= It)
14             Sn = GenerateNeighbor(Sc)
15             Sn.OF = CalculateOF(Sn)
16             IF (Sn.OF < Sc.OF) THEN
17                 CopySolution(Sc,Sn)
18                 IF (Sn.OF < Sb.OF) THEN
19                     CopySolution(Sb,Sn)
20                 ENDIF
21             ENDIF
22         ELSE
23              $\varphi = \text{random}(0,1)$ 
24              $\Delta = \text{Sn.OF} - \text{Sc.OF}$ 
25             IF ( $\varphi < e^{-\Delta/T}$ ) THEN
26                 CopySolution(Sc,Sn)
27             ENDIF
28         ENDELSE
29         Iter = Iter + 1
30     ENDWHILE
31     T =  $\alpha * T$ 
32 ENDWHILE
33 r = r + 1
34 T =  $T0 / e^{(\log(T0 / 100) / (Nr - r + 1))}$ 
35 T0 = T
36 CopySolution(Sc,Sb)
37 ENDWHILE

```

Fonte: Próprio autor.

Após a definição das ordens das pilhas e dos navios, a função *CreateSolution()* é acionada para gerar a solução inicial do problema. Esta função, apresentada detalhadamente no próximo subcapítulo, adota a seguinte estratégia: 1) seleciona a primeira pilha da ordem; 2) posiciona a pilha no diagrama dos pátios; 3) seleciona o navio referente à pilha alocada; 4) posiciona o navio no diagrama do cais, levando em conta a data de formação de sua pilha no pátio. Nas instâncias onde mais de uma pilha é carregada por navio, todas as pilhas referentes a este serão alocadas nos pátios antes que a função o selecione para atracação.

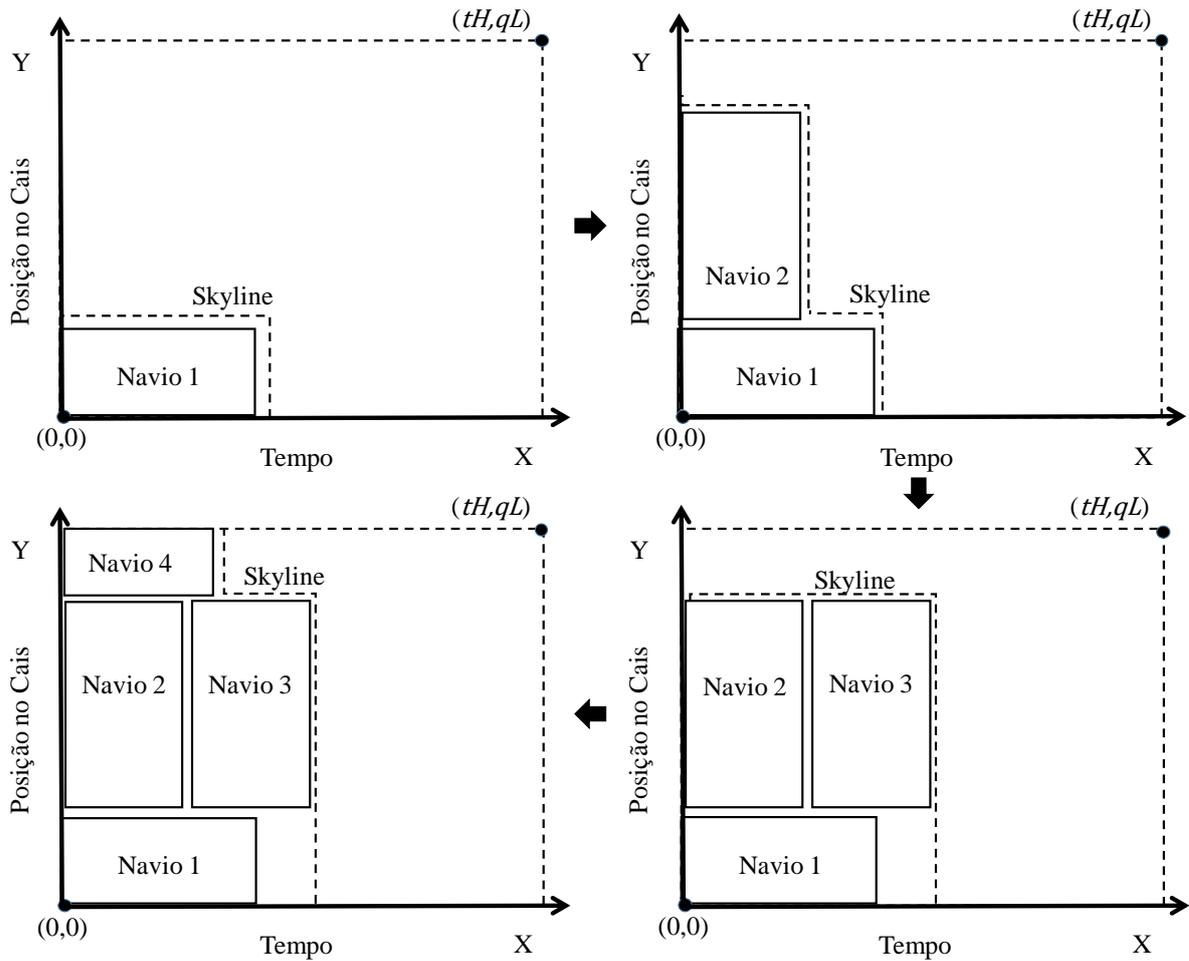
Para promover o arranjo das pilhas e dos navios em seus respectivos diagramas, é adotado o mesmo princípio: deve-se tentar posicionar o objetivo o mais próximo ao instante zero no eixo X (Tempo), ou seja, o mais cedo possível, respeitando as restrições impostas pelo problema.

O processo utilizado para determinar o arranjo das pilhas nos pátios é análogo ao executado para o arranjo dos navios no cais. A principal diferença entre estes se deve pela adaptação necessária em função da divisão entre os pátios, conforme explicado a seguir, iniciando pela lógica do posicionamento dos navios no diagrama do cais.

O arranjo dos navios começa a partir da coordenada (0,0), posicionando os navios, um após o outro, ao longo do eixo Y (Posição). Quando não for possível posicionar um navio, a função progredirá ao longo do eixo X (Tempo) e reiniciará o processo a partir do eixo Y. Para controlar a área disponível, uma variável denominada *skyline* irá traçar uma linha de contorno em torno dos navios já posicionados. A Figura 15 ilustra a execução da função e a utilização do *skyline*.

Quando o valor do *skyline* for inferior ao ETA do navio e/ou às datas de formação de suas respectivas pilhas no pátio, este valor será atualizado, avançando o *skyline* ao longo do diagrama, de modo à respeitar as premissas que impedem a atracação de um navio antes de sua chegada ao porto e da formação de suas pilhas.

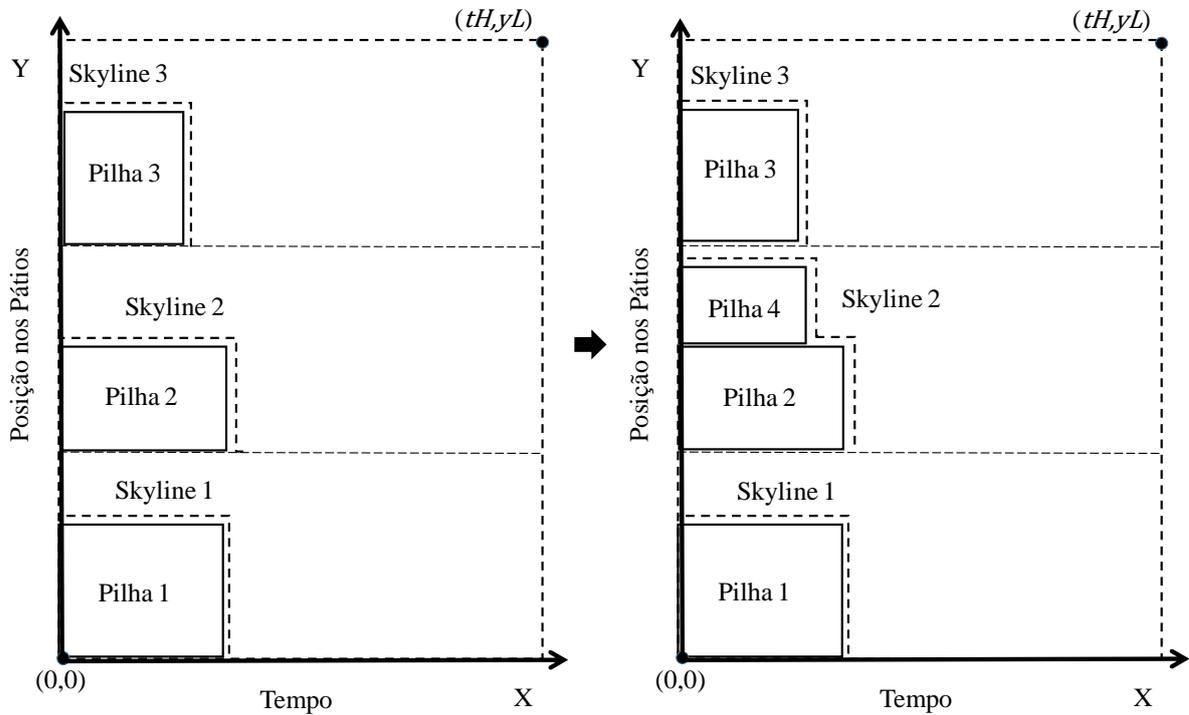
Figura 15 – Representação da função de arranjo dos navios



Fonte: Próprio autor.

No processo executado para o arranjo das pilhas, cada pátio contará com uma variável de *skyline* própria para controlar a disponibilidade de área. Considerando como exemplo uma instância com três pátios disponíveis, a alocação das pilhas começará a partir do Pátio 1, posicionando as pilhas ao longo do eixo Y. Caso não haja disponibilidade de área, passa-se para o próximo pátio, continuando o processo. A função só progredirá ao longo do eixo X após tentar posicionar a pilha nos três pátios. A Figura 16 ilustra este processo de posicionamento. Bem como no arranjo dos navios, as *skylines* dos pátios também são atualizadas, caso os seus valores sejam inferiores as datas de necessidade das pilhas a serem posicionadas.

Figura 16 – Representação da função de arranjo das pilhas



Fonte: Próprio autor

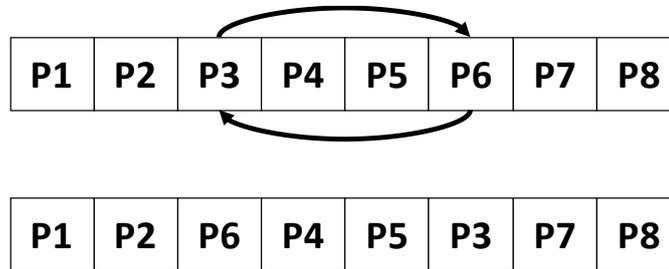
Com a conclusão dos arranjos das pilhas e dos navios, é obtida uma solução inicial completa. A solução atual  $Sc$  receberá esta solução gerada. Os vetores  $Sc.vet\_pilhas[n\_pil]$  e  $Sc.vet\_navios[n\_nav]$  são preenchidos, respectivamente, pela ordem utilizada pela função  $CreateSolution()$  para posicionar as pilhas e os navios em seus diagramas. Os vetores de estruturas  $Sc\_navios[n\_nav]$  e  $Sc\_pilhas[n\_pil]$ , definidos a partir das estruturas  $Sol\_Ship$  e  $Sol\_Pile$ , armazenam as coordenadas de cada um dos elementos.

O valor da função objetivo é calculado pela função  $CalculateOF(Sc)$ . A solução  $Sc$  é então copiada para a solução  $Sb$ , por meio da função  $CopySolution(Sb, Sc)$ , assumindo provisoriamente o status de melhor solução. Os vetores  $Sb.navios[n\_nav]$  e  $Sb.pilhas[n\_pil]$  também assumem os valores obtidos pela geração da solução inicial. Inicia-se, então, o processo iterativo que visa melhorar a solução obtida.

A cada iteração, uma solução vizinha é gerada por meio da função  $GenerateNeighbor(Sc)$ . Primeiro, esta função escolhe randomicamente entre cinco movimentos possíveis. Os dois primeiros movimentos envolvem modificar as posições

das pilhas no vetor  $Sn\_pilhas[n\_pil]$ . O Movimento 1, ilustrado na Figura 17, seleciona duas pilhas aleatórias e inverte as suas posições no vetor.

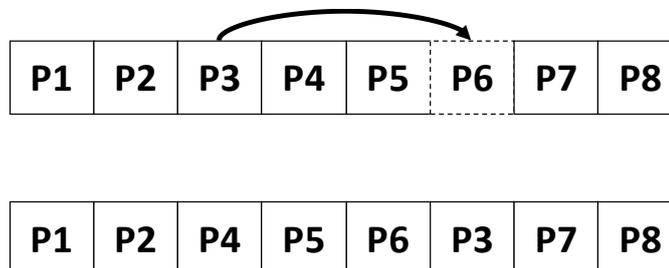
Figura 17 - Ilustração do Movimento 1



Fonte: Próprio autor

O Movimento 2 consiste em selecionar uma pilha qualquer e uma posição aleatória, diferente de sua posição atual, para qual ela será realocada. Conforme demonstrado pela Figura 18, ao executar este movimento, as pilhas presentes entre a posição antiga e a atual da pilha são afetadas, sendo deslocadas para frente ou para trás, ocupando a vacância deixada pela pilha sorteada.

Figura 18 – Ilustração do Movimento 2



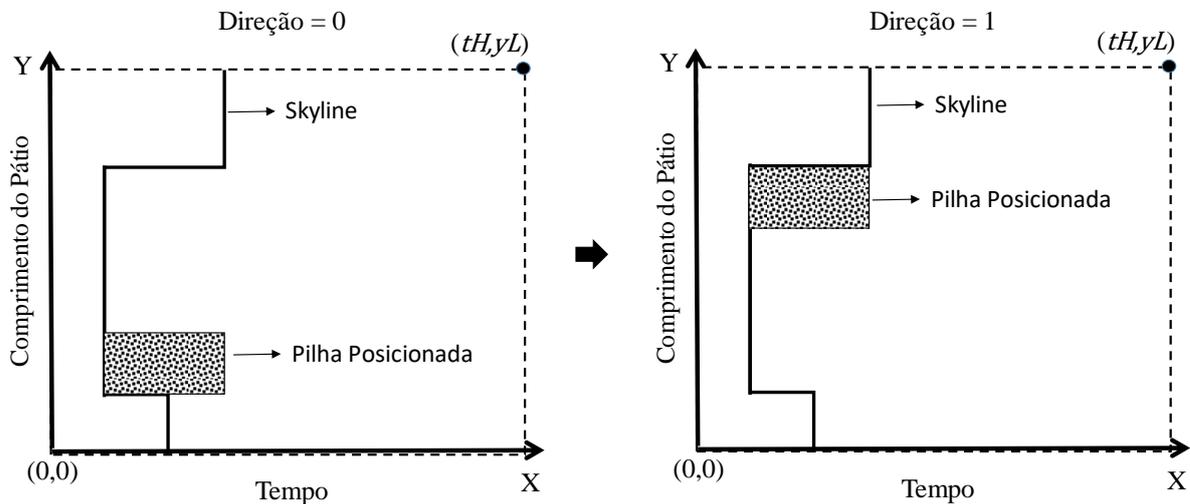
Fonte: Próprio autor

Os terceiro e quarto movimentos consistem em modificar o pátio de destino das pilhas. O Movimento 3 seleciona duas pilhas quaisquer e verifica se estas estão posicionadas em pátios diferentes. Caso tal condição se confirme, invertem-se os pátios de destino. O Movimento 4 consiste em selecionar apenas uma das pilhas e modificar o seu pátio de destino.

Para promover uma maior flexibilização no arranjo das pilhas no pátio, foi criado o movimento 5. Como foi descrito na seção anterior, no processo de posicionamento das pilhas, a função normalmente começa sua busca ao longo do eixo Y a partir da coordenada zero e vai subindo ao longo do pátio. O movimento 5 irá selecionar uma

pilha e inverterá a direção em relação ao eixo Y pela qual ela será posicionada no diagrama dos pátios. A Figura 19 ilustra a diferença entre o posicionamento normal (*direcao* = 0) e o posicionamento no sentido inverso (*direcao* = 1).

Figura 19 – Comparação entre o arranjo das pilhas em função do atributo *direcao*



Fonte: Próprio autor

Após a escolha de um dos movimentos descritos, a função *CreateSolution\_Sn()* é chamada para criar um novo arranjo das pilhas e dos navios. Esta função conta com uma lógica similar à função *CreateSolution()*. A diferença essencial é que em *CreateSolution()*, as pilhas não possuem um pátio definido para o seu posicionamento. Já na função *CreateSolution\_Sn()*, todas as pilhas possuem um pátio de origem determinado com base no vetor *Sn\_pilhas[n\_pil].patio*.

Após a geração de uma solução vizinha factível, uma nova função objetivo é calculada pela função *CalculateOF(Sn)*. Caso esta função objetivo seja melhor do que a atual, a solução vizinha é copiada para a solução atual por meio da função *CopySolution(Sc,Sn)*. Caso ela seja melhor do que a função objetivo da melhor solução vigente, ela é também copiada para este posto por meio da função *CopySolution(Sb,Sn)*.

Para evitar os mínimos locais, a meta-heurística proposta promove um teste para aceitar soluções inferiores à solução atual, gerando um número aleatório entre zero e 1,  $\varphi$ , e comparando o resultado com a fórmula  $e^{-\Delta/T}$ . Caso  $\varphi$  seja menor, a solução vizinha é copiada para a solução atual. Caso contrário, ela será descartada.

É proposto o reaquecimento da temperatura após a fase de resfriamento, visando promover uma busca local próxima a melhor solução encontrada. Para tal, o reaquecimento é feito com uma nova temperatura inicial menor, visando reduzir a probabilidade de aceitar soluções inferiores e sair do ótimo local. Assim, a nova temperatura inicial é calculada pela fórmula  $T = T0 / e^{(\log(T0 / 100) / (NR - r + 1))}$ , em que  $r$  representa a nova fase de reaquecimento,  $T$  é a nova temperatura inicial e  $T0$  é a temperatura inicial atual.

Na seção seguinte, as funções responsáveis pela construção das soluções são detalhadas para melhor compreensão do funcionamento da meta-heurística.

#### 5.4 DETALHAMENTO DAS FUNÇÕES

Nesta seção, a primeira função detalhada é a *CreateSolution()*, cujo pseudocódigo é apresentado na Figura 20. A função começa com a inicialização de seus parâmetros e variáveis, dentre estes, os valores de *skyline*. São considerados dois tipos de *skyline*, um skyline real (*skyline\_real*), que determina a área efetivamente ocupada pelas pilhas e um skyline auxiliar (*skyline\_aux*) que avança ao longo do diagrama de maneira independente na busca por uma posição viável para cada pilha.

Para definir quais posições deverão ser testadas para alocar cada pilha, são utilizadas as variáveis denominadas *xLAG*, *yLAG* e *patio*. Estas variáveis representam, respectivamente, as coordenadas no eixo X, no eixo Y e o pátio em si que está sendo analisado. Após a inicialização dos parâmetros, é dado início ao processo iterativo de posicionamento das pilhas. Na função *Reset\_Positions()*, a variável *xLAG* assume inicialmente o valor do atribuído ao *skyline\_aux* considerando o início do primeiro pátio. O *skyline\_aux* tem a princípio, o mesmo valor do *skyline\_real*, que delimita as posições que ainda não foram ocupadas.

Figura 20 – Pseudocódigo da função *CreateSolution*

```

1 InitializeParameters()
2 WHILE (j < num_pilhas)
3   nv = pilha[j].navio_carrega
4   Reset_Positions()
5   Search_Position()
6   Measure_Gap()
7   IF ((pilha[j].tamanho <= count) & (xLAG + pilha[j].formacao +
   pilha[j].navio_carrega <= h_tempo)) THEN
8     pilha[j].x = xLAG
9     pilha[j].y = yLAG
10    pilha[j].p = patio
11    Temp_Block_Yard()
12    Update_Skyline()
13    IF (xLAG + pilha[j].formacao > aux_atracacao[nv])
14      aux_atracacao[nv] = xLAG + pilha[j].formacao
15    ENDIF
16    quant_pilhas_alocadas[nv] = quant_pilhas_alocadas[nv] +1
17    IF (quant_pilhas_alocadas[nv] = max_pilhas_navio) THEN
18      Arrange_Ship()
19      Update_Yard_Utilization()
20      Update_Skyline()
21    ENDIF
22    j = j + 1
23  ENDIF
24  ELSE
25    Progress_Search()
26  ENDELSE
27 ENDWHILE

```

Fonte: Próprio autor.

As variáveis *yLAG* e *patio* também são modificadas em *Reset\_Positions()*, recebendo, respectivamente, os valores da coordenada inicial e do primeiro pátio. Então, a função *Search\_Position()* entra em um loop para identificar em qual posição, entre todos os pátios, encontra-se o menor valor de *skyline\_aux*. O pseudocódigo desta função é apresentado na Figura 21.

Figura 21 – Pseudocódigo da função *Search\_Position*

```

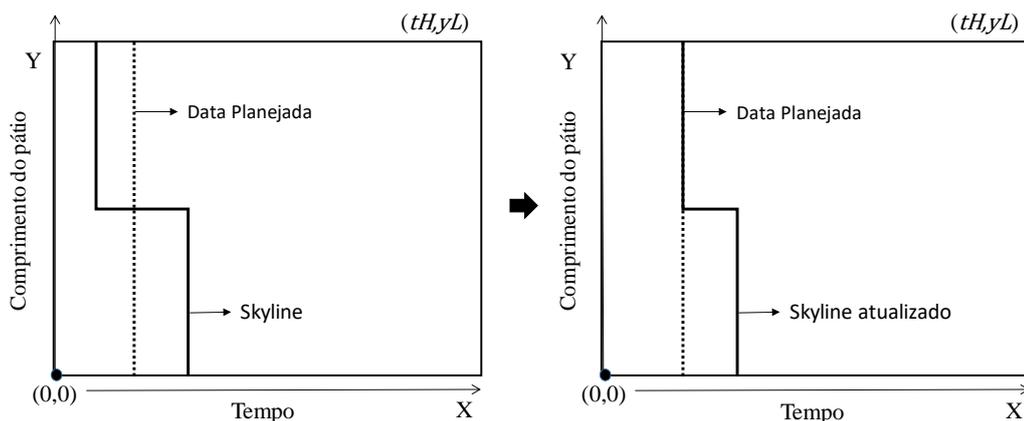
1  p = 1
2  WHILE (p ≤ num_pátios)
3      WHILE (i ≤ disp_pátio[p])
4          IF (skyline_aux[p][i] < pilha[j].data_necessidade) THEN
5              skyline_aux[p][i] < pilha[j].data_necessidade
6          ENDIF
7          IF (skyline_aux[p][i] < xLAG ) THEN
8              xLAG = skyline_aux[p][i]
9              yLAG = i
10             patio = p
11         ENDIF
12     ENDWHILE
13 ENDWHILE

```

Fonte: Próprio autor.

Ao longo das iterações, a função varre o diagrama dos pátios atualizando o valor do *skyline\_aux* em função da data planejada da pilha selecionada. Se o valor do skyline na coordenada selecionada for inferior ao valor da data planejada, o seu valor será igualado a ela, conforme ilustrado pela Figura 22. Ao identificar uma posição com valor de *skyline\_aux* menor que o *xLAG*, esta variável irá assumir o valor do skyline desta posição, *yLAG* irá assumir o valor da posição em si, representado por *i*, e *patio* irá assumir a identificação do pátio tratado.

Figura 22 – Atualização do skyline em função da data planejada da pilha



Fonte: Próprio autor

Após a identificação da posição de menor valor do *skyline*, é feita uma busca para descobrir qual o comprimento de cais disponível para atracação, considerando a coordenada no X identificada por *xLAG*. A função *Measure\_Gap()*, cujo pseudocódigo é apresentado na Figura 23, é responsável por identificar a dimensão desta lacuna.

Iniciando-se na posição identificada por  $yLAG$ , a função avança no diagrama acrescentando valor à um contador auxiliar ( $count$ ) enquanto o valor de  $skyline\_aux$  naquele ponto for igual ao  $xLAG$ .

Figura 23 - Pseudocódigo da função *Measure\_Gap*

```

1 count = 0
2 i = yLAG
3 WHILE (i ≤ disp_patio[patio]) & (skyline_aux[patio][i] == xLAG)
4     count = count + 1
5     i = i + 1
6 ENDWHILE

```

Fonte: Próprio autor.

Partindo da coordenada identificada pelas variáveis  $xLAG$ ,  $yLAG$  e  $patio$ , a função irá verificar a possibilidade de alocar a pilha nesta posição. Em termos de comprimento, é feita a comparação entre o tamanho da pilha e o valor do contador auxiliar. Se o tamanho da pilha for menor ou igual, significa que, no que tange ao eixo Y, é possível aloca-la nessa posição.

Em termos de tempo, é verificado se a soma do  $xLAG$  com os tempos de formação da pilha e carregamento do seu respectivo navio é inferior ao horizonte de tempo. Se esta condição também for satisfeita, a pilha será alocada neste pátio e nesta posição, com a coordenada x sendo igualada ao  $xLAG$  e a coordenada y igualada ao  $yLAG$ .

Após o posicionamento da pilha, é feita a atualização das variáveis de *skyline*. Ao longo da posição do pátio ocupada pela pilha, a função *Temp\_Block\_Yard()* fará com que o *skyline\_real* assuma o valor equivalente ao horizonte de tempo  $h\_tempo$ . Isto é feito para evitar uma possível sobreposição por outra pilha, visto que, enquanto não for definido o momento de atracação de seu respectivo navio, não se sabe por quanto tempo a pilha armazenada permanecerá no pátio. O pseudocódigo desta função é apresentado na Figura 24.

Figura 24 - Pseudocódigo da função *Temp\_Block\_Yard*

```

1 i = 0
2 WHILE (i < pilha[j].tamanho)
3     skyline_real[patio][ylag + i] = h_tempo
4     i = i + 1
5 ENDWHILE

```

Fonte: Próprio autor.

Posteriormente, a função *Update\_Skyline()* é chamada para igualar o valor de *skyline\_aux* ao valor de *skyline\_real* ao longo do diagrama. Após posicionar a(s) pilha(s) de um navio, este deverá ser posicionado no diagrama do cais. A função *Arrange\_Ship()*, responsável pelo posicionamento dos navios e que será comentada ao longo desta seção, é chamada toda vez que suas pilhas forem posicionadas. Uma das variáveis de entrada desta função é o *aux\_atracacao*, variável auxiliar que captura o momento em que as pilhas de um navio tiveram sua formação concluída no pátio. Esta variável é utilizada para assegurar a premissa de que os navios só podem atracar no cais a partir do instante em que suas pilhas já estiverem prontas para o carregamento.

Após posicionar um navio em seu diagrama, é definido o instante em que o carregamento de suas pilhas é encerrado. Assim, sabe-se o momento em que a área de pátio previamente ocupada por sua(s) pilha(s) é liberada. Com isto, o *skyline\_real*, que havia sido igualado ao valor de *h\_tempo* na posição da pilha, pode ser recalculado para assumir o valor real em que a pilha não ocupa mais o pátio. A função responsável por esse processo é *Update\_Yard\_Utilization()*, apresentada na Figura 25, que identifica as posições ocupadas por todas as pilhas do navio atracado e corrige o seu *skyline*. Posteriormente, a função *Update\_Skyline()* é novamente chamada para igualar o *skyline\_aux* ao *skyline\_real*.

Figura 25 - Pseudocódigo da função *Update\_Yard\_Utilization*

```

1  j_aux = j
2  i = 0
3  WHILE (i < max_pilhas_navio)
4      IF (pilha[j_aux].navio_carrega = nv) THEN
5          pilha[j_aux].t_espera = navio[nv].x - (pilha[j_aux].x +
              navio[nv].operacao)
6          k = pilha[j_aux].y
7          p = pilha[j_aux].patio
8          WHILE (k < pilha[j_aux].tamanho + pilha[j_aux].y)
9              skyline_real[p][k] = navio[nv].x + navio[nv].operacao
10             k = k + 1
11         ENDWHILE
12         i = i + 1
13     ENDIF
14     j_aux = j_aux - 1
15 ENDWHILE

```

Fonte: Próprio autor.

Caso as condições para que a pilha seja alocada não tenham sido satisfeitas, a função deverá atualizar o *skyline\_aux* no espaço analisado antes de reiniciar a busca, preenchendo a lacuna analisada. Esta tarefa é realizada pela função *Progress\_Search()*, apresentada na Figura 26. O espaço a ser preenchido vai desde a posição representada pela variável *yLAG* até *yLAG + count*. O valor do skyline nas posições anterior e posterior ao espaço no diagrama são observados e o menor entre estes é utilizado para atualizar o *skyline\_aux*, ocupando a lacuna demarcada.

Figura 26 - Pseudocódigo da função *Progress\_Search*

```

1 IF (skyline[patio][yLAG - 1] <= skyline_aux[patio][y_LAG + count + 1])
2     neighbour = skyline_aux[patio][yLAG - 1]
3 ENDIF
4 ELSE
5     neighbour = skyline_aux[patio][y_LAG + count + 1]
6 ENDELSE
7 i = yLAG
8 WHILE (i < yLAG + count)
9     skyline_aux[patio][i] = neighbour
10    i = i + 1
11 ENDWHILE

```

Fonte: Próprio autor.

Após o término das iterações, a função haverá alocado todas as pilhas no diagrama dos pátios, concluindo esta porção da solução inicial. Em paralelo, a medida em que a função de arranjo dos navios é chamada, define-se também os momentos e as posições de atracação de todos os navios na instância. Esta função, denominada *Arrange\_Ship()*, é explicada a seguir.

A função *Arrange\_Ship()*, cujo pseudocódigo é descrito na Figura 27, segue uma lógica análoga ao arranjo das pilhas. Um ponto que diferencia esta função é a utilização da variável denominada *Flag*. Ao contrário do processo de arranjo das pilhas, promovido pela função *CreateSolution()*, apenas um navio será alocado quando a função for chamada. O processo iterativo então se baseará nesta *Flag* para encerrar o processo iterativo de busca, o que ocorrerá quando for determinada uma posição viável para atracação do navio.

Figura 27 - Pseudocódigo da função *Arrange\_Ship*

```

1 InitializeParameters()
2 Flag = 0
3 WHILE (Flag = 0)
4     Reset_Positions()
5     Search_Position()
6     Measure_Gap()
7     IF (navio[nv].comprimento <= count) THEN
8         navio[nv].x = xLAG
9         navio[nv].y = yLAG
10        navio[nv].berco = berco
11        Update_Skyline()
12        Flag = 1
13    ENDIF
14    ELSE
15        Progress_Search()
16    ENDELSE
17 ENDWHILE

```

Fonte: Próprio autor.

A função se inicia com a inicialização dos parâmetros utilizados no processo de busca. A variável *xLAG* irá assumir o valor do *skyline\_aux* no início do cais, ou do primeiro berço em caso de berços discretos, e a função entrará no processo iterativo que visa identificar a melhor posição de atracação para o navio.

A função *Search\_Position()* utilizada para varrer o cais opera de modo similar à sua função homônima do diagrama dos pátios. O valor de *skyline\_aux* também é modificado neste loop, levando em conta, entretanto a variável *aux\_atracacao[nv]* calculada em *CreateSolution()*. Esta variável garante que o posicionamento dos navios respeitará a premissa de que estes não podem ser arranjados em seu diagrama antes do seu ETA, nem antes da formação de suas pilhas no pátio.

Com a utilização da função *Measure\_Gap()*, determina-se a dimensão da lacuna a ser testada para atracação do navio. A função *Arrange\_Ship()* irá então avaliar a viabilidade de se alocar do navio na posição determinada. Se as condições forem satisfeitas, este será posicionado no diagrama do cais, assumindo as coordenadas determinadas por *xLAG* e *yLAG*. Caso contrário, a função *Progress\_Search()* será chamada para atualizará o *skyline\_aux* preenchendo a lacuna testada, e a busca por uma posição viável seguirá.

Ao término do posicionamento do navio selecionado, a função *Arrange\_Ship()* se encerra e a meta-heurística retorna para função *CreateSolution()*, recebendo a informação do momento em que o navio irá desatracar, o que permite determinar o momento em que as posições ocupadas pelas suas pilhas serão liberadas. Ao término desta função, é obtida uma solução completa, com todas as pilhas e navios alocados em seus respectivos diagramas.

## 6 RESULTADOS E ANÁLISES

Este capítulo apresenta os resultados obtidos pela meta-heurística nos experimentos computacionais desenvolvidos nesta pesquisa. Primeiramente, é descrita a etapa de calibração dos parâmetros e, posteriormente, é feita uma análise dos resultados obtidos nas instâncias desenvolvidas por Bordal (2020).

### 6.1 CALIBRAÇÃO DOS PARÂMETROS

Nesta etapa, foi promovida a calibração dos seguintes parâmetros: temperatura inicial ( $T_0$ ), coeficiente de resfriamento ( $\alpha$ ) e número de iterações para cada temperatura ( $It$ ). Os valores dos parâmetros temperatura de congelamento ( $fT$ ) e número de reaquecimentos ( $Nr$ ) foram definidos previamente, sendo fixados em 0,01 e 3, respectivamente.

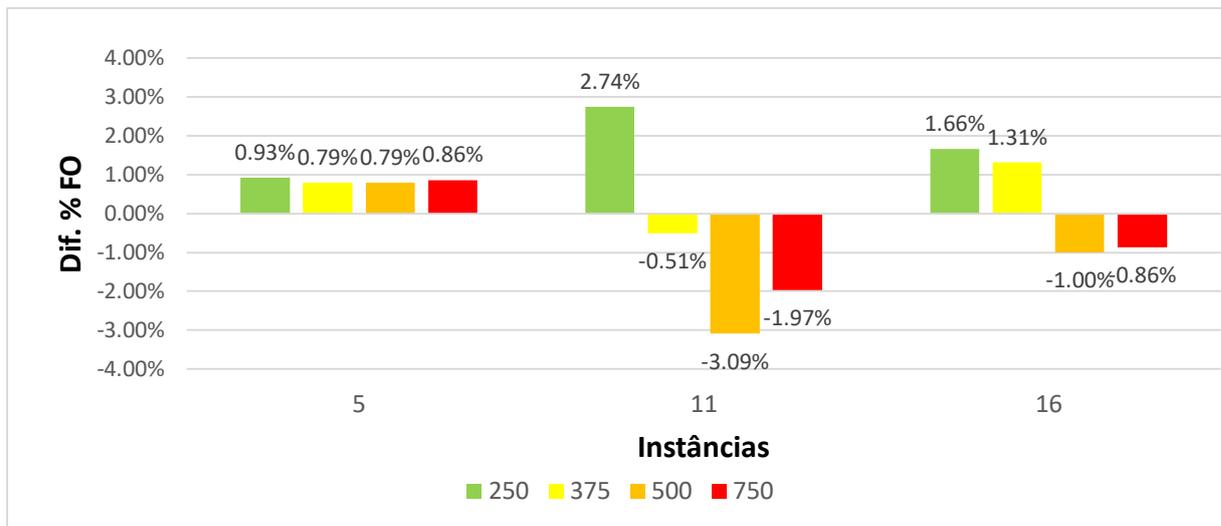
O processo de calibração dos parâmetros foi realizado a partir do seguinte procedimento: para cada parâmetro analisado, foram definidas 4 configurações de parâmetros utilizados na execução da meta-heurística, em cada configuração, o valor do parâmetro analisado foi variado, enquanto os demais foram mantidos fixos, permitindo analisar isoladamente a influência do valor de cada parâmetro no desempenho do algoritmo.

Para estes testes, foram definidas três instâncias para serem utilizadas: as instâncias 5, 11 e 16. Com cada uma destas instâncias, a meta-heurística é executada 5 vezes para cada configuração dos parâmetros. Os resultados das cinco execuções são extraídos e é feita a média das funções objetivos obtidas. Esta média é comparada com o resultado obtido pelo modelo para descobrir a diferença percentual entre a FO média da meta-heurística e a FO do CPLEX.

A diferença percentual é calculada subtraindo o valor da FO do CPLEX da FO média da meta-heurística e dividindo o seu resultado pela FO do CPLEX. Quando a diferença percentual for positiva, ela indicará que o resultado obtido pela execução do modelo pelo CPLEX. Quando a diferença percentual for negativa, significa que a meta-heurística obteve um resultado médio superior ao CPLEX.

O primeiro parâmetro avaliado foi a temperatura inicial  $T_0$ . Para a sua calibração,  $T_0$  assumiu os seguintes valores: 250, 375, 500 e 500. Os valores de  $\alpha$  e  $It$  foram fixados em 0,97 e 75, respectivamente. A Figura 28 mostra os resultados dos testes realizados com diferentes valores para o parâmetro  $T_0$ . O eixo X, denominado *Dif. % FO*, indica a diferença percentual da FO entre as execuções da meta-heurística e do CPLEX. O eixo Y mostra as instâncias testadas, com 4 barras para cada instância, onde cada barra representa um valor de  $T_0$  testado.

Figura 28 – Resultado dos testes de calibração da temperatura inicial



Fonte: Próprio autor

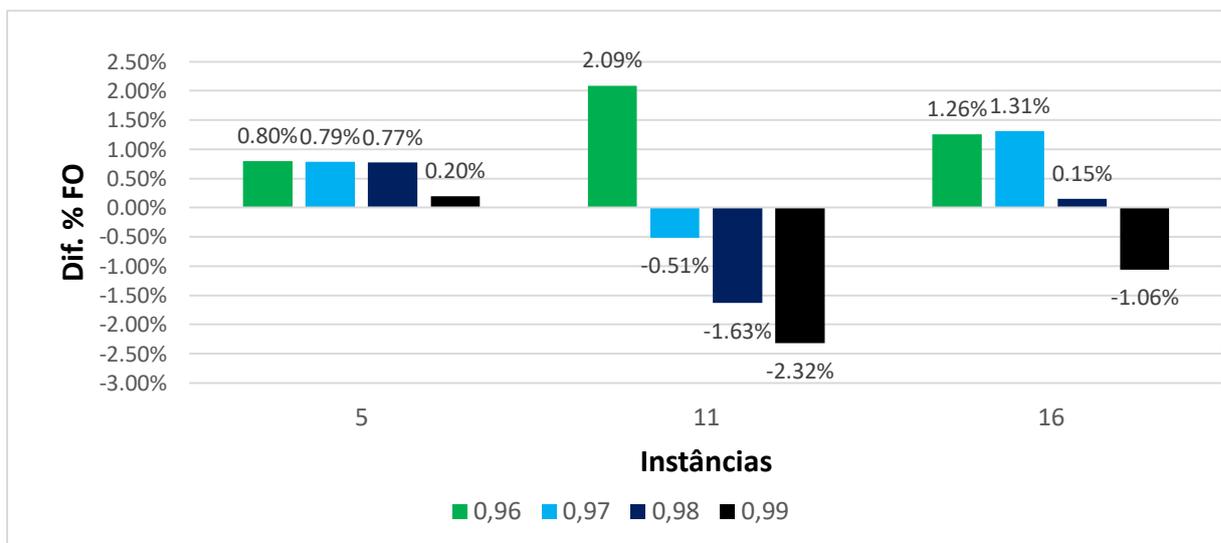
Ao analisar o gráfico, é possível observar que os resultados obtidos nos testes com  $T_0$  igual a 250 (cor verde) obteve os piores resultados, em especial nas instâncias 11 e 16, definidas como instâncias de planejamento de médio prazo de acordo com a descrição de Bordal (2020). Este resultado se justifica na medida em que uma menor temperatura inicial implica na redução do número de iterações realizadas pela meta-heurística, o que significa que menos combinações de soluções foram analisadas. Além disso, ao deixar de realizar iterações com temperaturas acima de 250, perde-se a possibilidade de se aceitar movimentos que a priori pioram a solução, mas que permitem ao algoritmo explorar novas combinações dentro do espaço de busca, evitando ótimos locais.

Ao aumentar o valor de  $T_0$  para 375 (cor amarela), pode-se notar a evolução das soluções obtidas, refletidas em uma menor diferença percentual da FO nas instâncias analisadas. Este progresso permaneceu nas instâncias 11 e 16 ao estabelecer o valor de 500 (cor laranja) para a temperatura inicial, atingindo os menores valores em termos de diferença percentual da FO.

Ao definir  $T_0$  com um valor de 750 (cor vermelha), o valor da diferença percentual sofreu um acréscimo em relação aos testes com temperatura inicial de 500. Este comportamento se embasa na ampliação da probabilidade de aceite de soluções inferiores pela meta-heurística nas iterações com uma temperatura mais elevada, distanciando a solução corrente das melhores soluções encontradas.

O segundo parâmetro analisado foi o coeficiente de resfriamento  $\alpha$ . Para a sua calibração,  $\alpha$  foi testado com os seguintes valores: 0,96; 0,97; 0,98 e 0,99. Os valores de  $T_0$  e  $It$  foram fixados em 375 e 75, respectivamente. A Figura 29 mostra os resultados dos testes realizados com os diferentes valores de  $\alpha$ . O eixo X indica a diferença percentual da FO entre as execuções da meta-heurística e do CPLEX. O eixo Y mostra as instâncias testadas, também com 4 barras para cada instância, onde cada barra representa um valor de  $\alpha$  testado.

Figura 29 – Resultado dos testes de calibração do coeficiente de resfriamento



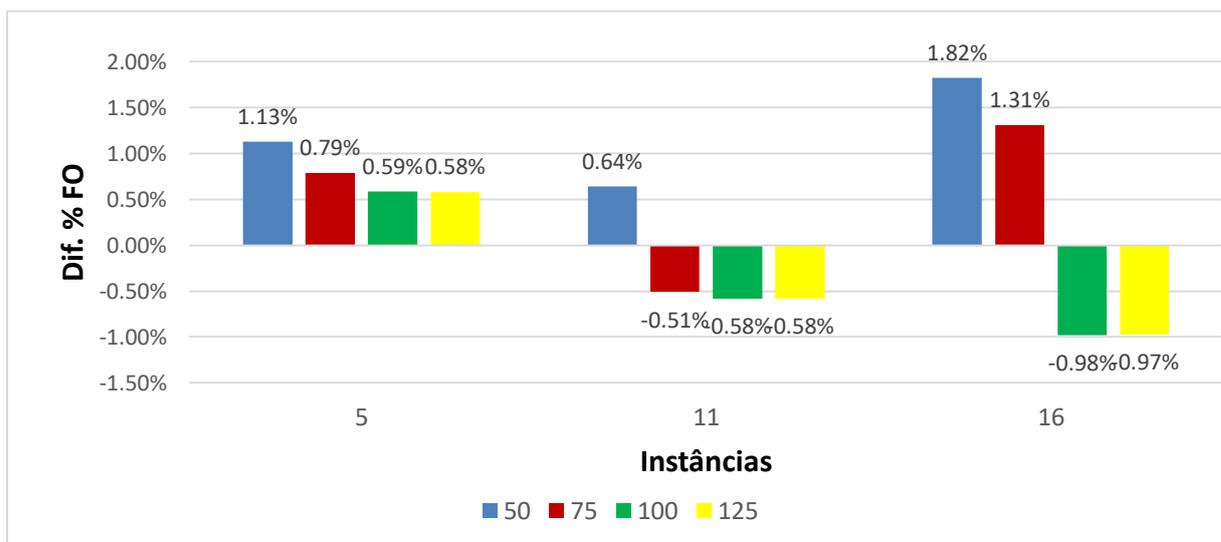
Fonte: Próprio autor

A análise do gráfico indica claramente a influência do coeficiente de resfriamento no desempenho da meta-heurística em termos de diferença percentual. Os testes

realizados com  $\alpha$  definido em 0,96 e 0,97 obtiveram os piores resultados nas instâncias analisadas. Ao progredir com o valor de  $\alpha$ , aumentou-se a qualidade das soluções obtidas, principalmente nas instâncias 11 e 16, referentes ao planejamento de longo prazo. Os melhores resultados foram obtidos pelos testes realizados com  $\alpha$  assumindo o valor de 0,99.

O último parâmetro analisado foi o número de iterações para cada temperatura  $It$ . Para calibrar o  $It$ , foram feitos testes assumindo os seguintes valores: 50, 75, 100 e 125. Os valores de  $T0$  e  $\alpha$  foram fixados em 375 e 0,97, respectivamente. A Figura 30 mostra os resultados dos testes realizados com os diferentes valores de  $It$ . Novamente, o eixo X indica a diferença percentual da FO e o eixo Y mostra as instâncias testadas, com as barras representando os diferentes valores de  $It$  testados.

Figura 30 – Resultado dos testes de calibração do número de iterações a cada temperatura



Fonte: Próprio autor

Nos experimentos promovidos, a execução das instâncias de teste com  $It$  fixado em 50 iterações obteve os piores resultados. Ao aumentar o número de iterações para 75 e, posteriormente, para 100, a diferença percentual da FO foi reduzida, atestando a superioridade das soluções encontradas sob estes parâmetros pela meta-heurística. Entretanto, ao subir o número de iterações a cada temperatura para 125, não houve melhora significativa da FO nos testes. Diante disto, buscando aprimorar o desempenho da meta-heurística considerando também o aspecto do tempo de

execução, julga-se 100 como um valor apropriado para o número de iterações a cada temperatura.

Concluída a calibração dos parâmetros, determinou-se que os seguintes valores seriam utilizados: Temperatura inicial ( $T_0$ ) = 500; Coeficiente de resfriamento ( $\alpha$ ) = 0,99 e; Número de iterações para cada temperatura ( $It$ ) = 100. Como explicitado previamente, os valores da temperatura de congelamento ( $fT$ ) e do número de reaquecimentos ( $Nr$ ) foram fixados em 0,01 e 3, respectivamente. Na seção a seguir, são apresentados os resultados obtidos pela meta-heurística proposta nas instâncias desenvolvidas por Bordal (2020), comparando-os com os resultados obtidos pelo modelo matemático.

## 6.2 RESULTADOS E ANÁLISES.

O modelo matemático desenvolvido por Bordal (2020) foi resolvido por meio do solver IBM® ILOG® CPLEX® Versão 12.8, utilizando um computador com dois processadores Intel Xeon Silver 4114, de 12 núcleos de 2.10 GHz, e 128 GB de memória RAM. Nos seus experimentos com o modelo matemático, Bordal (2020) estabeleceu um tempo limite para execução do CPLEX de 10.800 segundos, o equivalente a 3 horas.

A meta-heurística proposta nesta pesquisa foi desenvolvida em linguagem C e executada em um computador com dois processadores Intel Xeon E5-2640 v3, com 8 núcleos de 2.60 GHz, e 64 GB de memória RAM. Para melhor avaliar o desempenho da meta-heurística, cada instância foi executada 10 vezes, permitindo verificar a sua estabilidade.

Na Tabela 3, são apresentados os resultados obtidos em cada instância pela meta-heurística e pelo modelo matemático executado no CPLEX. A coluna 1 indica o número da instância. Nas colunas de 2 a 6, são apresentadas as informações relativas ao desempenho da meta-heurística. A coluna *FO Média* apresenta a média da função objetivo nas 10 execuções de cada instância. A coluna *Desv. Padrão Relativo FO* indica o desvio padrão relativo da Função objetivo, calculado a partir do desvio padrão das FOs obtidas, dividido pela média das FOs. A coluna *Melhor FO* apresenta o menor entre os valores da FO obtido em todas as execuções. A coluna *Tempo Exec. (s)*

mostra (em segundos) a média de tempo necessário para concluir a execução da meta-heurística. Por fim, a coluna *Tempo p/ Melhor (s)* apresenta a média de tempo (em segundos) necessária para encontrar a melhor solução em cada execução.

Tabela 3 – Resultados obtidos pela meta-heurística e pelo CPLEX nas instâncias testadas

Instância	Meta-Heurística					CPLEX				Comparação		
	FO Média	Desv. Padrão Relativo FO	Melhor FO	Tempo Exec. (s)	Tempo P/ Melhor (s)	FO	LB	UB	GAP	Tempo Exec. (s)	Dif % FO	Dif. Tempo Total (s)
1	2170,00	0,00%	2170	30,25	2,27	2133	-	-	0,00%	8072	1,73%	-8041,75
2	1470,00	0,00%	1470	32,20	2,44	1470	-	-	0,00%	75	0,00%	-42,80
3	1802,00	0,00%	1802	31,58	1,92	1802	-	-	0,00%	36	0,00%	-4,42
4	1270,00	0,00%	1270	35,04	0,92	1270	-	-	0,00%	25	0,00%	10,04
5	1294,30	0,12%	1293	36,54	20,95	1297	1101	1297	15,11%	10800	-0,21%	-10763,46
6	832,00	0,12%	823	38,83	16,73	822	759	822	7,66%	10800	1,22%	-10761,17
7	1101,00	0,00%	1101	39,20	4,08	1101	-	-	0,00%	1015	0,00%	-975,80
8	759,00	0,00%	759	41,68	2,20	759	-	-	0,00%	327	0,00%	-285,32
9	18088,80	0,27%	18019	133,76	74,16	18732	5215	18732	72,16%	10800	-3,43%	-10666,24
10	6430,00	0,80%	6375	155,55	83,42	6404	3667	6404	42,74%	10800	0,41%	-10644,45
11	12750,30	0,86%	12561	130,68	79,68	13149	4207	13149	68,01%	10800	-3,03%	-10669,32
12	5365,50	0,13%	5361	146,49	82,45	6208	2491	6208	59,88%	10800	-13,57%	-10653,51
13	4397,00	0,59%	4373	149,31	95,25	4319	2759,5	4319	36,11%	10800	1,81%	-10650,69
14	2767,40	0,26%	2754	155,11	64,84	2799	2616,2	2799	6,53%	10800	-1,13%	-10644,89
15	3991,30	0,19%	3982	166,30	95,24	4058	1834,4	4058	54,79%	10800	-1,64%	-10633,70
16	2683,30	0,20%	2678	169,40	94,36	2798	2618,2	2798	6,43%	10800	-4,10%	-10630,60
17	2785,00	0,00%	2785	26,48	1,86	2770	-	-	0,00%	266	0,54%	-239,52
18	3563,00	0,00%	3563	24,64	2,64	3563	-	-	0,00%	2	0,00%	22,64
19	4237,00	0,00%	4237	24,65	0,97	4212	-	-	0,00%	2	0,59%	22,65
20	11252,00	0,00%	11252	26,70	6,58	11252	-	-	0,00%	124	0,00%	-97,30
21	16082,00	0,00%	16082	24,84	2,62	16082	-	-	0,00%	2	0,00%	22,84
22	20068,00	0,00%	20068	24,96	0,34	19818	-	-	0,00%	1	1,26%	23,96
23	1814,00	0,00%	1814	26,14	3,23	1814	-	-	0,00%	7	0,00%	19,14
24	2172,00	0,00%	2172	26,98	2,20	2172	-	-	0,00%	3	0,00%	23,98
25	2478,00	0,00%	2478	24,92	0,78	2478	-	-	0,00%	2	0,00%	22,92

Fonte: Próprio autor

As colunas de 7 a 11 apresentam informações do desempenho do CPLEX. A coluna *FO* contém a função objetivo encontrada pelo modelo. As colunas *LB* e *UB* são, respectivamente, o *Lower Bound* e o *Upper Bound* da solução, sendo exibidos quando o CPLEX encontra não encontra a solução ótima dentro do tempo limite de execução. A coluna *GAP* é calculada a partir da fórmula  $(UB-LB)/UB$ . A coluna *Tempo Exec. (s)*, apresenta o tempo de execução do CPLEX. As colunas 12 e 13 apresentam as

comparações entre os resultados da meta-heurística proposta e do CPLEX. A coluna *Dif. % FO* refere-se à diferença percentual média entre as funções objetivo obtidas, calculado como: a diferença entre a FO Média da meta-heurística e a FO do CPLEX, dividida pela FO do CPLEX. Quando o valor resultante for positivo, significa que o CPLEX apresentou a melhor solução. Caso o valor seja negativo, a meta-heurística foi capaz de encontrar uma solução melhor em comparação à resolução do CPLEX. A coluna *Dif. Tempo Total(s)* apresenta a diferença real (em segundos) entre o tempo médio de execução da meta-heurística e o tempo de execução do CPLEX.

Analisando os resultados apresentados, é possível perceber a estabilidade da meta-heurística. Em 15 das 25 instâncias obteve-se um desvio padrão relativo de 0%, o que significa que as 10 execuções encontraram o mesmo valor para a função objetivo. Em 7 instâncias o desvio padrão relativo ficou abaixo de 0,5%. Apenas os testes nas instâncias 10, 11 e 13 resultaram em maiores desvios, variando de 0,56% a 0,86%.

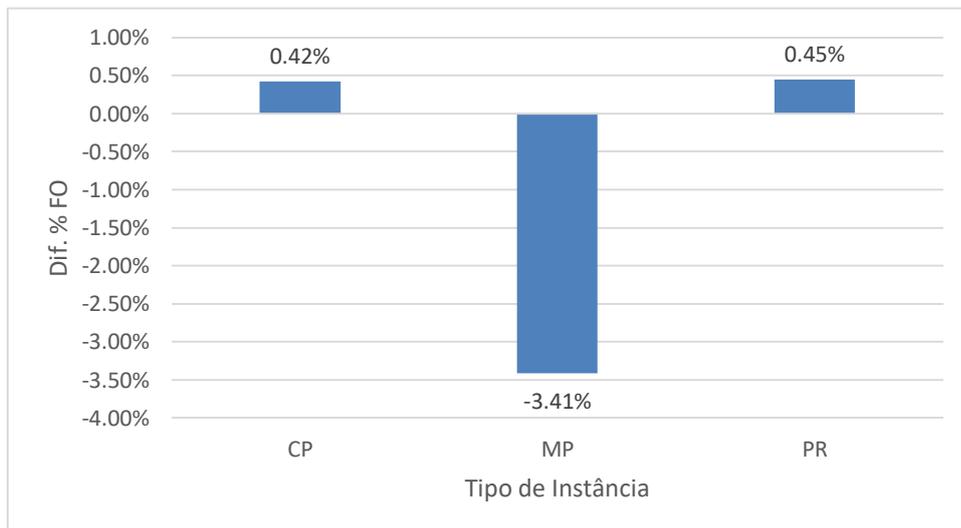
Comparando diretamente com os resultados encontrados pela execução do modelo matemático, identifica-se que a meta-heurística encontrou a solução ótima em 11 instâncias. Em 7 instâncias, nas quais o CPLEX não foi capaz de encontrar a solução ótima dentro do tempo limite de execução, a meta-heurística foi capaz de obter soluções superiores, com destaque para a instância 12, onde houve uma notável melhoria de 13,57% em termos de diferença percentual da FO. Dentre as instâncias onde a meta-heurística não obteve uma solução igual ou superior ao modelo matemático, a maior diferença percentual foi obtida na instância 13 com um valor de 1,81%.

Para ilustrar melhor o comparativo entre a meta-heurística e o CPLEX, as instâncias foram agrupadas em três categorias e as médias das diferenças percentuais de FO foram analisadas. As categorias definidas foram: planejamento de curto prazo (CP), planejamento de médio prazo (MP) e instâncias com pátio de disponibilidade reduzida (PR). As instâncias do tipo CP se referem a todas as instâncias dos grupos 1 e 2, que contam com 10 navios a serem atendidos e pelo menos um pátio com 100% de sua capacidade disponível. As instâncias do tipo MP se referem a todas as instâncias dos grupos 3 e 4, com um total de 42 navios para serem carregados. As instâncias que compõem os grupos 5, 6 e 7, que contam com pátios parcialmente indisponíveis,

foram colocadas em conjunto na categoria PR, separadas das demais instâncias com 10 navios.

A Figura 31 apresenta a média da diferença percentual entre a média das FOs obtidas pela meta-heurística e os resultados encontrados pelo CPLEX em cada categoria de instâncias. Em ambas as categorias de instâncias com 10 navios (CP e PR), o CPLEX obteve melhores resultados, porém a diferença percentual média da FO foi inferior à 0,5%. Nas instâncias de planejamento de médio prazo, a diferença percentual encontrada foi de -3,41%, atestando uma vantagem significativa a favor da meta-heurística.

Figura 31 – Diferenças percentuais da FO em cada categoria de instâncias

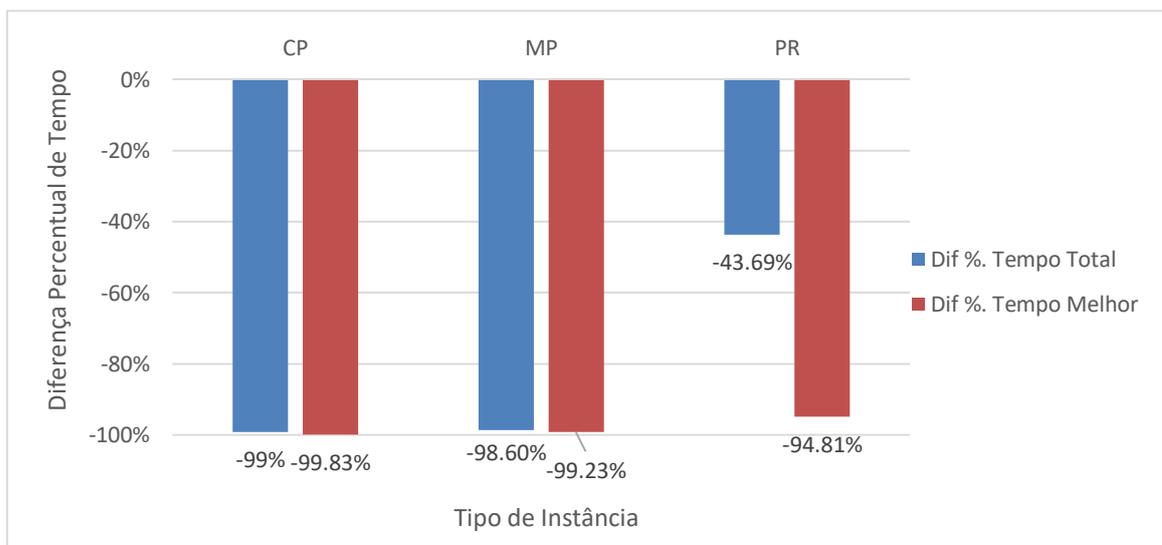


Fonte: Próprio autor

Tratando-se dos tempos de execução das instâncias, a utilização da meta-heurística proposta é considerada significativamente favorável em relação à execução do modelo no CPLEX. Em média, a meta-heurística necessita de 69s para executar todas as instâncias testadas, enquanto o CPLEX, com tempo limitado à 3h para cada instância, obteve uma média de 4718s por execução. Em 17 das 25 instâncias, a meta-heurística se mostrou mais rápida em relação ao CPLEX, chegando a uma diferença superior à 2h em 11 destas instâncias. Nas instâncias em que o modelo matemático encontrou a solução em menor tempo, a diferença foi inferior à 30s.

Considerando novamente a separação das instâncias em três categorias, a Figura 32 apresenta a comparação dos tempos de execução em termos de diferença percentual. As informações apresentadas como *Dif % Tempo Total* consideram a diferença do tempo médio de execução total da meta-heurística em relação ao CPLEX, enquanto *Dif % Tempo Melhor* leva em consideração o tempo médio para achar a melhor solução. Com uma diferença percentual próxima aos -100%, é possível inferir que diferença no tempo de execução nas instâncias do tipo CP e MP é extremamente considerável em relação ao CPLEX. Para as instâncias de pátio reduzido, a diferença no tempo total de execução não tem a mesma proporção, entretanto, considerando os tempos médios para obtenção das melhores soluções, nota-se que a meta-heurística converge rapidamente, atingindo o seu ápice com uma diferença percentual de -94,81% em relação à execução do CPLEX.

Figura 32 – Diferenças percentuais de tempo de execução em cada categoria de instâncias

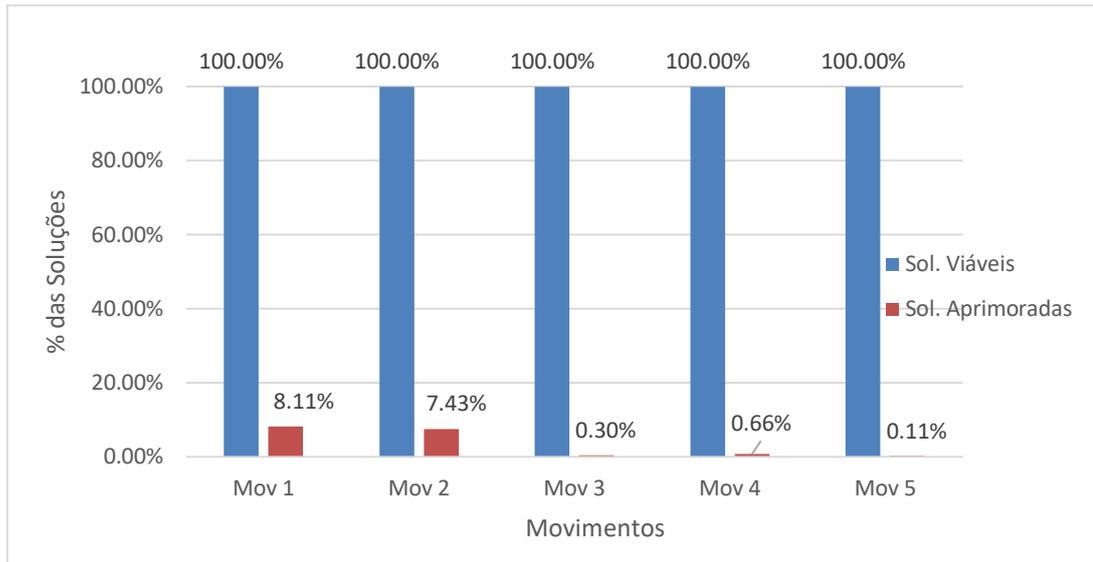


Fonte: Próprio autor

Para demonstrar o comportamento da meta-heurística, as instâncias 4, 11 e 16 foram selecionadas para uma análise de suas execuções. Primeiramente, os movimentos serão analisados para demonstrar com qual frequência eles resultaram em soluções viáveis e com qual frequência eles melhoraram a solução vigente da meta-heurística durante a execução. A Figura 33 apresenta as porcentagens de soluções viáveis e das soluções aprimoradas por cada um dos movimentos na instância. Todos os movimentos promovidos geraram soluções viáveis. Um comportamento semelhante a este foi observado em todas as instâncias dos Grupos 1 e 2. Em termos de melhoria

da solução atual, nota-se que os movimentos 1 e 2 tiveram uma influência consideravelmente maior para o avanço da solução obtida em relação aos demais movimentos.

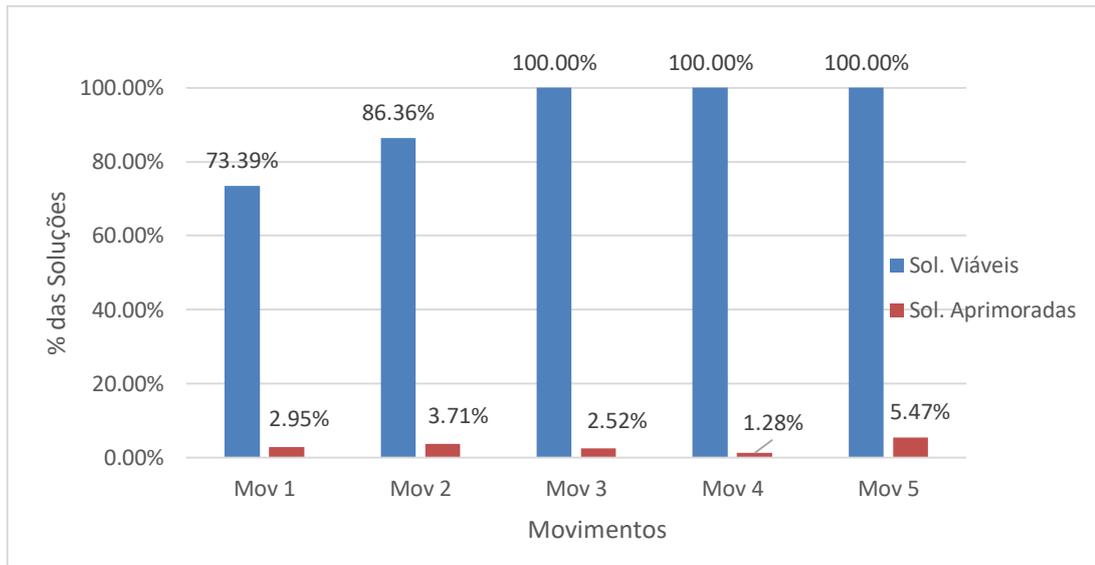
Figura 33 – Proporção de soluções viáveis e aprimoradas pelos movimentos na Instância 4



Fonte: Próprio autor

Na Figura 34, são apresentadas porcentagens de soluções viáveis e das soluções aprimoradas por cada movimento na instância 11. Ao contrário da instância 4, os movimentos 1 e 2 geraram algumas soluções inviáveis. Além disso, a proporção de movimentos responsáveis por promover melhorias na solução foi alterada. O movimento 5, criado para flexibilizar a alocação das pilhas no pátio, foi o principal responsável por estes aprimoramentos.

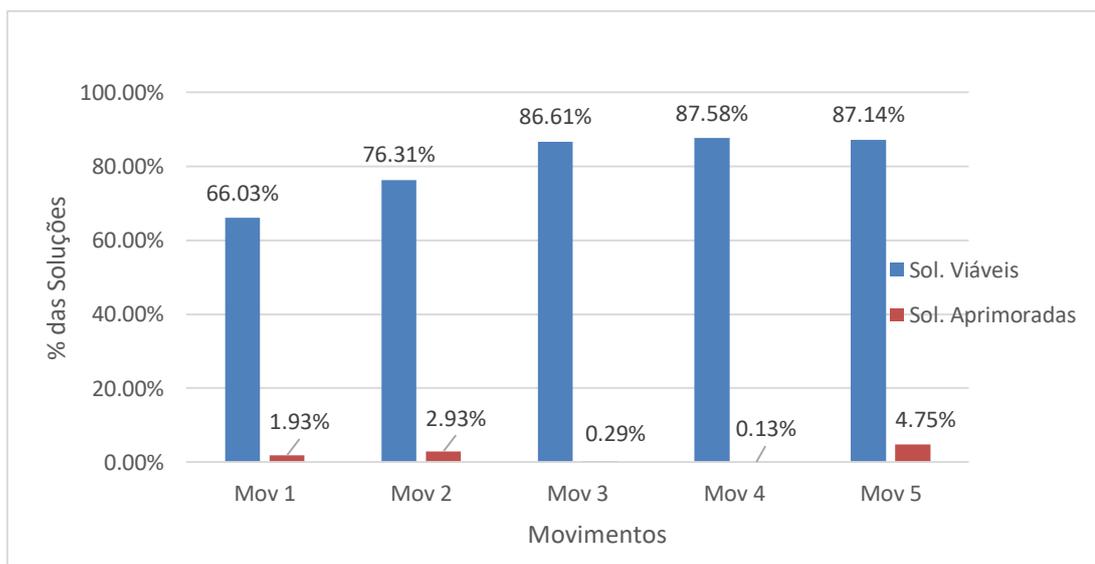
Figura 34 – Proporção de soluções viáveis e aprimoradas pelos movimentos na Instância 11



Fonte: Próprio autor

A Figura 35 apresenta a porcentagem de soluções viáveis e das soluções aprimoradas por movimento na instância 16. Nesta instância, nota-se o aumento de solução inviáveis geradas, com destaque para o movimento 1 que gerou uma solução viável em 66,03% das vezes que foi acionado. No que tange à melhoria das soluções obtidas por meio dos movimentos, o movimento 5 se mostrou novamente útil ao ser responsável pela maioria dos movimentos positivos da meta-heurística.

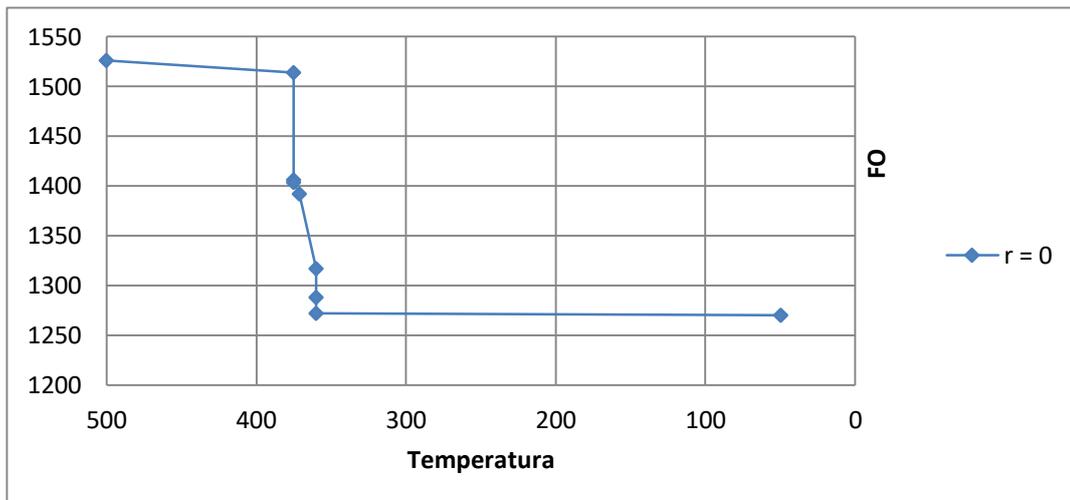
Figura 35 – Proporção de soluções viáveis e aprimoradas pelos movimentos na Instância 16



Fonte: Próprio autor

Além da questão dos movimentos, a convergência da função objetivo ao longo do decréscimo da temperatura e das rodadas de reaquecimento foi analisada. A Figura 36 apresenta a convergência da FO na instância 4. O eixo X representa a temperatura, que decresce ao decorrer a da meta-heurística, e o eixo Y indica o valor da FO. Assim como ocorreu na maioria das instâncias de 10 navios, a meta-heurística obteve o seu melhor valor da FO relativamente rápido, ainda na primeira rodada de reaquecimento.

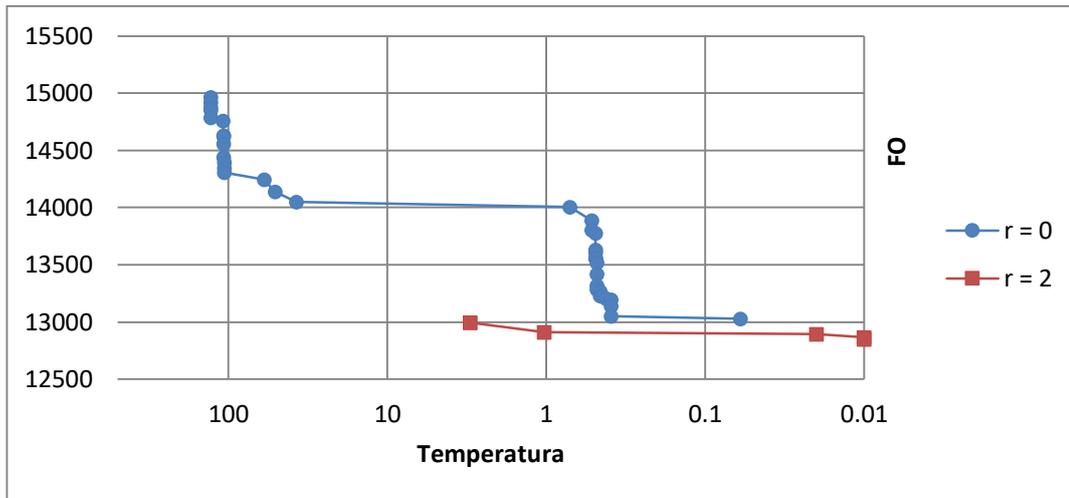
Figura 36 – Convergência da FO ao longo da meta-heurística na Instância 4



Fonte: Próprio autor

A Figura 37 apresenta a convergência da FO na instância 11. Em função da concentração de pontos, que indicam melhorias sucessivas da FO em temperaturas semelhantes, perto do ponto de congelamento, o eixo X foi representado em escala logarítmica. Como ilustrado pelo gráfico, desta vez a segunda rodada de reaquecimento foi fundamental para contribuir com a melhoria da função objetivo, embasando a utilização das rodadas de reaquecimento.

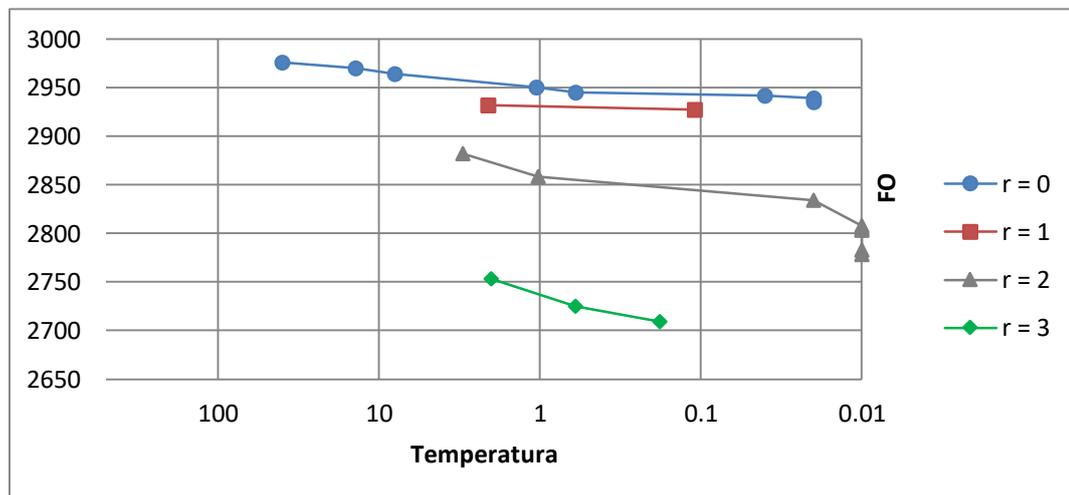
Figura 37 – Convergência da FO ao longo da meta-heurística na Instância 11



Fonte: Próprio autor

Na Figura 38, é apresentada a convergência da FO na instância 16. Assim como na instância 11, o eixo X foi representado em escala logarítmica em função da concentração de pontos com a temperatura próximo de 0. Como é possível notar, todas as rodadas de reaquecimento foram fundamentais para a convergência da FO até a solução final encontrada.

Figura 38 – Convergência da FO ao longo da meta-heurística na Instância 16



Fonte: Próprio autor

Analisando o aspecto da convergência da função objetivo ao longo das iterações da meta-heurística, nota-se que a complexidade da instância influi diretamente no tempo necessário para a encontrar a sua melhor solução. Entretanto, destaca-se que, mesmo nas maiores instâncias, o tempo de execução foi inferior à 3 minutos. Mesmo

com tempos de execução significativamente inferiores aos obtidos pela resolução do modelo matemático por meio do CPLEX, os piores resultados em termos de diferença percentual da FO não foram superiores à 2%.

De fato, a meta-heurística obteve, em média, soluções melhores nas instâncias de planejamento de médio prazo em relação ao CPLEX. A qualidade das soluções geradas, aliada ao menor tempo de execução, demonstra a validade da meta-heurística desenvolvida para auxiliar no planejamento integrado da fila de navios e da designação de áreas de pátio para estocagem de pelotas em um porto graneleiro, permitindo não somente projetar um plano para um período pré-estabelecido, mas também replanejar e propor diversos cenários, apropriando-se da capacidade da meta-heurística de gerar soluções em períodos aceitáveis para este fim.

## 7 CONCLUSÕES

Esta pesquisa abordou o problema do planejamento da fila de navios integrado à designação de áreas do pátio de armazenagem de granel para atender ao plano de produção de uma usina de pelotização, visando minimizar a diferença entre as datas planejadas para formação das pilhas de pelotas e as datas efetivas, além de minimizar o tempo de permanência dos navios no porto.

O problema em questão foi tratado por Bordal (2020), que apresentou um modelo matemático para formular o problema, resolvendo as instâncias desenvolvidas por meio do CPLEX. Dada a natureza do problema e os próprios resultados obtidos pela formulação matemática, foi identificada a importância do desenvolvimento de métodos heurísticos capazes de gerar soluções de boa qualidade em tempo computacional hábil para aplicação prática. Assim, foi proposta neste estudo uma meta-heurística baseada no *Simulated Annealing* (SA) para resolver este problema de planejamento integrado.

A meta-heurística desenvolvida foi avaliada com base nos resultados obtidos pelo modelo matemático desenvolvido por Bordal (2020). Nas instâncias de pequeno porte, referentes ao planejamento de curto prazo, a meta-heurística obteve em média uma diferença percentual da FO inferior à 0,5% em relação ao CPLEX, indicando a proximidade dos resultados obtidos com as soluções ótimas geradas. Nas instâncias classificadas como de planejamento de médio prazo, a meta-heurística foi capaz de gerar soluções superiores ao CPLEX, atingindo em média uma diferença percentual de -3,41%.

Em termos de tempo de execução, a meta-heurística foi capaz de resolver todas as instâncias em menos de três minutos, enquanto a resolução do modelo matemático por meio do CPLEX atingiu o tempo limite de execução, delimitado em três horas. A diferença significativa entre os tempos de execução em relação ao CPLEX, em conjunto com a qualidade das soluções encontradas, comprovou o desempenho satisfatório do algoritmo desenvolvido.

Considerando os resultados obtidos, é possível atestar a validade da meta-heurística proposta para adaptação e utilização como uma ferramenta de apoio ao processo de planejamento integrado da fila de navios no porto e da designação de áreas dos pátios

de armazenagem para formação das pilhas de pelota. Os baixos tempos de execução permitem, inclusive, a sua utilização para replanejar a produção e para gerar e avaliar diferentes cenários pertinentes à operação do porto.

Como recomendação de trabalhos futuros, é incentivado o desenvolvimento de novas meta-heurísticas, baseadas em outras abordagens, para aprimorar os resultados obtidos nas instâncias do problema. Além disso, propõem-se a inserção de novas restrições ao modelo matemático e, conseqüentemente, as meta-heurísticas desenvolvidas em conjunto, para incluir aspectos relacionados à programação das empilhadeiras/recuperadoras no pátio e à capacidade limitada dos carregadores de navio, resultando em uma formulação mais fidedigna à realidade da operação portuária.

## REFERÊNCIAS

AGRA, A.; OLIVEIRA, M. MIP approaches for the integrated berth allocation and quay crane assignment and scheduling problem. **European Journal of Operational Research**, v. 264, n. 1, p. 138–148, jan. 2018.

AL-HAMMADI, J.; DIABAT, A. An integrated berth allocation and yard assignment problem for bulk ports: Formulation and case study. **RAIRO Operations Research**, v. 51, n. 1, p. 267–284, jan-mar 2017.

BABU, S. A. I. *et al.* Minimizing delay of ships in bulk terminals by simultaneous ship scheduling stockyard planning and train scheduling. **Maritime Economics & Logistics**, v. 17, n. 4, p. 464–492, dez 2015.

BARROS, V. H. *et al.* Model and heuristic for berth allocation in tidal bulk ports with stock level constraints. **Computers & Industrial Engineering**, v. 60, n. 4, p. 606–613, maio 2011.

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, maio 2010.

BIERWIRTH, C.; MEISEL, F. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. **European Journal of Operational Research**, v. 244, n. 3, p. 675–689, ago. 2015.

BLOODSHED. **IDE Bloodshed Dev C++ Version 5.11**. Disponível em: <<https://www.bloodshed.net/dev/>>. Acesso em: 14 jan. 2019.

BOLAND, N.; GULCZYNSKI, D.; SAVELSBERGH, M. A stockyard planning problem. **EURO Journal on Transportation and Logistics**, v. 1, n. 3, p. 197–236, set 2012.

BORDAL, T. **Planejamento da alocação de áreas de um pátio de granel mineral integrado ao atendimento de navios no porto para atender a produção de uma usina de pelletização**. Dissertação de Mestrado. Programa de Pós-Graduação em Engenharia Civil. Universidade Federal do Espírito Santo, Vitória, 2020.

CORDEAU, J.-F. *et al.* Models and Tabu Search Heuristics for the Berth-Allocation Problem. **Transportation Science**, v. 39, n. 4, p. 526–538, nov. 2005.

CORRECHER, J. F.; ALVAREZ-VALDES, R. A biased random-key genetic algorithm for the time-invariant berth allocation and quay crane assignment problem. **Expert Systems with Applications**, v. 89, p. 112–128, dez. 2017.

DNPM - DEPARTAMENTO NACIONAL DE PRODUÇÃO MINERAL. **Informe Mineral 1º/2018**. Brasília, 2018. Disponível em: <<http://www.anm.gov.br/dnpm/publicacoes/serie-estatisticas-e-economia-mineral/informe-mineral/publicacoes-nacionais>>. Acesso em: 06 jan. 2019.

ERNST, A. T. *et al.* Mathematical models for the berth allocation problem in dry bulk terminals. **Journal of Scheduling**, v. 20, n. 5, p. 459–473, 16 out. 2017.

HANSEN, P.; OĞUZ, C.; MLADENOVIĆ, N. Variable neighborhood search for minimum cost berth allocation. **European Journal of Operational Research**, v. 191, n. 3, p. 636–649, dez. 2008.

HE, J. Berth allocation and quay crane assignment in a container terminal for the trade-off between time-saving and energy-saving. **Advanced Engineering Informatics**, v. 30, n. 3, p. 390–405, ago. 2016.

HENDRIKS, M. P. M.; LEFEBER, E.; UDDING, J. T. Simultaneous berth allocation and yard planning at tactical level. **OR Spectrum**, v. 35, p. 441–456, mar 2013.

HU, Q.-M.; HU, Z.-H.; DU, Y. Berth and quay-crane allocation problem considering fuel consumption and emissions from vessels. **Computers & Industrial Engineering**, v. 70, p. 1–10, abr. 2014.

HU, Z.-H. Multi-objective genetic algorithm for berth allocation problem considering daytime preference. **Computers & Industrial Engineering**, v. 89, p. 2–14, nov. 2015.

IMAI, A. *et al.* Berth allocation in a container port: using a continuous location space approach. **Transportation Research Part B: Methodological**, v. 39, n. 3, p. 199–221, mar. 2005.

IMAI, A. *et al.* Berth allocation at indented berths for mega-containerships. **European Journal of Operational Research**, v. 179, n. 2, p. 579–593, jun. 2007.

IMAI, A. *et al.* The simultaneous berth and quay crane allocation problem. **Transportation Research Part E: Logistics and Transportation Review**, v. 44, n. 5, p. 900–920, set. 2008.

IMAI, A.; NISHIMURA, E.; PAPADIMITRIOU, S. The dynamic berth allocation problem for a container port. **Transportation Research Part B: Methodological**, v. 35, n. 4, p. 401–417, maio 2001.

IMAI, A.; NISHIMURA, E.; PAPADIMITRIOU, S. Berth allocation with service priority. **Transportation Research Part B: Methodological**, v. 37, n. 5, p. 437–457, jun. 2003.

IMAI, A.; NISHIMURA, E.; PAPADIMITRIOU, S. Berthing ships at a multi-user container terminal with a limited quay capacity. **Transportation Research Part E: Logistics and Transportation Review**, v. 44, n. 1, p. 136–151, jan. 2008.

IMAI, A.; YAMAKAWA, Y.; HUANG, K. The strategic berth template problem. **Transportation Research Part E: Logistics and Transportation Review**, v. 72, p. 77–100, dez. 2014.

JIN, J. G.; LEE, D. H.; HU, H. Tactical berth and yard template design at container transshipment terminals: A column generation based approach. **Transportation Research Part E: Logistics and Transportation Review**, v. 73, p. 168–184, jan 2015.

KIRKPATRICK S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **Science**, v. 220, n. 4598, p. 671–680, mai 1983.

KORDIĆ, S. *et al.* Combinatorial approach to exactly solving discrete and hybrid berth allocation problem. **Applied Mathematical Modelling**, v. 40, n. 21–22, p. 8952–8973, nov. 2016.

LALLA-RUIZ, E. *et al.* A Set-Partitioning-based model for the Berth Allocation Problem under Time-Dependent Limitations. **European Journal of Operational Research**, v. 250, n. 3, p. 1001–1012, maio 2016.

LEE, D. H.; JIN, J. G. Feeder vessel management at container transshipment terminals. **Transportation Research Part E: Logistics and Transportation Review**, v. 49, p. 201–216, jan 2013.

LIANG, C.; HUANG, Y.; YANG, Y. A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. **Computers & Industrial Engineering**, v. 56, n. 3, p. 1021–1028, abr. 2009.

LIM, A. The berth planning problem. **Operations Research Letters**, v. 22, n. 2–3, p. 105–110, mar. 1998.

LIN, S.W.; TING, C.J.; WU, K.C. Simulated annealing with different vessel assignment strategies for the continuous berth allocation problem. **Flexible Services and Manufacturing Journal**, v. 30, n. 4, p. 740–763, 14 dez. 2018.

LIU, M. *et al.* Bi-objective optimization for the container terminal integrated planning. **Transportation Research Part B: Methodological**, v. 93, p. 720–749, nov 2016.

LUZ, A. B.; SAMPAIO, J. A.; FRANÇA, S. C. A. **Tratamento de Minérios**. 5ª ed. Rio de Janeiro: Centro de Tecnologia Mineral - CETEM, 2010.

MA, H. L. *et al.* An integrated model for berth and yard planning in container terminals with multi-continuous berth layout. **Annals of Operations Research**, v. 273, p. 409–431, fev 2019.

MAURI, G. R. *et al.* An adaptive large neighborhood search for the discrete and continuous Berth allocation problem. **Computers & Operations Research**, v. 70, p. 140–154, jun. 2016.

MAZIOLI, F. C. *et al.* Assessment of the impact of Charter Party clauses and port's characteristics on the port's financial result. **Computers & Industrial Engineering**, v. 128, p. 70–90, fev. 2019.

METROPOLIS N. *et al.* Equation of state calculations by fast computing machines. **The Journal of Chemical Physics**, v. 21, n. 6, p. 1087–1092, jun 1953.

NISHIMURA, E.; IMAI, A.; PAPADIMITRIOU, S. Berth allocation planning in the public berth system by genetic algorithms. **European Journal of Operational Research**, v. 131, n. 2, p. 282–292, jun. 2001.

OLIVEIRA, R. M. DE; MAURI, G. R.; LORENA, L. A. N. Clustering Search for the Berth Allocation Problem. **Expert Systems with Applications**, v. 39, n. 5, p. 5499–5505, abr. 2012.

PARK, Y.-M.; KIM, K. H. A scheduling method for Berth and Quay cranes. **OR Spectrum**, v. 25, n. 1, p. 1–23, 1 fev. 2003.

QIN, T.; DU, Y.; SHA, M. Evaluating the solution performance of IP and CP for berth allocation with time-varying water depth. **Transportation Research Part E: Logistics and Transportation Review**, v. 87, p. 167–185, mar. 2016.

RIBEIRO, G. M. *et al.* Berth allocation in an ore terminal with demurrage, despatch and maintenance. **Computers & Industrial Engineering**, v. 96, p. 8–15, jun. 2016.

ROBENEK, T. *et al.* A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports. **European Journal of Operational Research**, v. 235, n. 2, p. 399–411, jun. 2014.

ŞAHİN, C.; KUVVETLİ, Y. Differential evolution based meta-heuristic algorithm for dynamic continuous berth allocation problem. **Applied Mathematical Modelling**, v. 40, n. 23–24, p. 10679–10688, dez. 2016.

SAMARCO. **Relatório anual de sustentabilidade 2014**. Belo Horizonte, 2015. Disponível em: <<https://www.samarco.com/relatorios/>>. Acesso em: 8 jan. 2019.

SAMARCO. Disponível em: <<https://www.samarco.com/produto/>>. Acesso em: 4 jan. 2019.

TAO, Y.; LEE, C. Y. Joint planning of berth and yard allocation in transshipment terminals using multi-cluster stacking strategy. **Transportation Research Part E: Logistics and Transportation Review**, v. 83, p. 34–50, nov 2015.

TING, C.-J.; WU, K.-C.; CHOU, H. Particle swarm optimization algorithm for the berth allocation problem. **Expert Systems with Applications**, v. 41, n. 4, p. 1543–1550, mar. 2014.

TÜRKOĞULLARI, Y. B. *et al.* Optimal berth allocation and time-invariant quay crane assignment in container terminals. **European Journal of Operational Research**, v. 235, n. 1, p. 88–101, maio 2014.

UMANG, N.; BIERLAIRE, M.; VACCA, I. Exact and heuristic methods to solve the berth allocation problem in bulk ports. **Transportation Research Part E: Logistics and Transportation Review**, v. 54, p. 14–31, ago. 2013.

UNSAL, O.; OGUZ, C. An exact algorithm for integrated planning of operations in dry bulk terminals. **Transportation Research Part E: Logistics and Transportation Review**, v. 126, p. 103–121, jun 2019.

URSAVAS, E.; ZHU, S. X. Optimal policies for the berth allocation problem under stochastic nature. **European Journal of Operational Research**, v. 255, n. 2, p. 380–387, dez. 2016.

USGS - U.S. GEOLOGICAL SURVEY. **Mineral Commodity Summaries 2018**. 2018. Disponível em: <<https://minerals.usgs.gov/minerals/pubs/mcs/>>. Acesso em: 12 jan. 2019.

VENTURINI, G. *et al.* The multi-port berth allocation problem with speed optimization and emission considerations. **Transportation Research Part D: Transport and Environment**, v. 54, p. 142–159, jul. 2017.

WANG, K. *et al.* Column generation for the integrated berth allocation, quay crane assignment, and yard assignment problem. **Transportation Science**, v. 52, n. 4, p. 812–834, jun. 2018.

XIANG, X.; LIU, C.; MIAO, L. A bi-objective robust model for berth allocation scheduling under uncertainty. **Transportation Research Part E: Logistics and Transportation Review**, v. 106, p. 294–319, out. 2017.

XIANG, X.; LIU, C.; MIAO, L. Reactive strategy for discrete berth allocation and quay crane assignment problems under uncertainty. **Computers & Industrial Engineering**, v. 126, p. 196–216, dez. 2018.

XU, D.; LI, C.-L.; LEUNG, J. Y.-T. Berth allocation with time-dependent physical limitations on vessels. **European Journal of Operational Research**, v. 216, n. 1, p. 47–56, jan. 2012.

YAN, S. *et al.* A network flow model for the dynamic and flexible berth allocation problem. **Computers & Industrial Engineering**, v. 81, p. 65–77, mar. 2015.

ZHEN, L.; CHANG, D.-F. A bi-objective model for robust berth allocation scheduling. **Computers & Industrial Engineering**, v. 63, n. 1, p. 262–273, ago. 2012.

ZHEN, L.; CHEW, E. P.; LEE, L. H. An integrated model for berth template and yard template planning in transshipment hubs. **Transportation Science**, v. 45, n. 4, p. 483–504, nov. 2011.

ZHEN, L.; LEE, L. H.; CHEW, E. P. A decision model for berth allocation under uncertainty. **European Journal of Operational Research**, v. 212, n. 1, p. 54–68, jul. 2011.