



BRUNO LUÍS HÖNIGMANN CERESER

UM MODELO MATEMÁTICO PARA OTIMIZAR O
DESCARREGAMENTO DE NAVIOS NUM TERMINAL
GRANELEIRO

CAMPINAS
2013



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística
e Computação Científica

BRUNO LUÍS HÖNIGMANN CERESER

UM MODELO MATEMÁTICO PARA OTIMIZAR O
DESCARREGAMENTO DE NAVIOS NUM TERMINAL
GRANELEIRO

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em matemática aplicada.

Orientador: Luiz Leduíno de Salles Neto

Coorientador: Antonio Carlos Moretti

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO ALUNO BRUNO LUÍS HÖNIGMANN CERESER, E ORIENTADA PELO PROF. DR. LUIZ LEDUÍNO DE SALLES NETO.

Assinatura do Orientador

Luiz Leduíno de Salles Neto

Assinatura do Coorientador

Antonio Carlos Moretti

CAMPINAS
2013

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

C334m Cereser, Bruno Luís Hönigmann, 1988-
Um modelo matemático para otimizar o descarregamento de navios num terminal graneleiro / Bruno Luís Hönigmann Cereser. – Campinas, SP : [s.n.], 2013.

Orientador: Luiz Leduino de Salles Neto.

Coorientador: Antonio Carlos Moretti.

Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Problema de alocação de berços. 2. Alocação de máquinas portuárias. 3. Controle de tráfego marítimo - Modelos matemáticos. I. Salles Neto, Luiz Leduino de. II. Moretti, Antonio Carlos, 1958-. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: A mathematical model to optimize the unloading of ships in a grain terminal

Palavras-chave em inglês:

Berth allocation problem

Port machinery allocation

Maritime traffic control - Mathematical models

Área de concentração: Matemática Aplicada

Titulação: Mestre em Matemática Aplicada

Banca examinadora:

Luiz Leduino de Salles Neto [Orientador]

Aurelio Ribeiro Leite de Oliveira

Anibal Tavares de Azevedo

Data de defesa: 30-09-2013

Programa de Pós-Graduação: Matemática Aplicada

Dissertação de Mestrado defendida em 30 de setembro de 2013 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

L. de d. Lduino

Prof.(a). Dr(a). LUIZ LEDUÍNO DE SALLES NETO

Aurelio R. de Oliveira

Prof.(a). Dr(a). AURELIO RIBEIRO LEITE DE OLIVEIRA

A. T. Azevedo

Prof.(a). Dr(a). ANIBAL TAVARES DE AZEVEDO

Abstract

The maritime transportation is essential to global companies. In 2009 80% of global trade was made by maritime transportation, and over the year this number has raised.

Because of this high flux, the BAP (Berth Allocation Problem) arose. The BAP have intent to discovery the vessel allocation sequence in every berth with the minimum total wait time.

In this work, the berths of BAP will be considered dependents, we also considered the division of portuary machines between consecutive berths. The objective of this work is the mathematical representation of the terminals Produtos Diversos and Praia Mole of the Tubarão port. For this reasons we propose here unprecedented mathematical models.

Keywords: Berth Allocation Problem, Port Machinery Allocation, Dependents berths.

Resumo

O transporte marítimo é fundamental para grande empresas de caráter global. Em 2009 80% do comércio mundial foi realizado via transporte marítimo e desde então esse número só tem crescido.

Devido a esse alto fluxo nos portos mundiais, surgiu o problema de alocação de navios em berços, que tem por finalidade estabelecer uma sequência de atendimento de navios para cada berço disponível no cais, visando minimizar o tempo total da operação e espera dos navios dentro dos portos.

Nesse trabalho o problema de alocação de navios em berços será abordado considerando berços dependentes, e também a divisão das máquinas portuárias em berços consecutivos aplicados a terminais do Porto de Tubarão (ES). Para a representação matemática dos processos desses terminais do porto de tubarão foram desenvolvidos novos modelos matemáticos. Os testes computacionais foram exitosos.

Palavras-chave: Problema de Alocação de Berços, Alocação de Máquinas Portuárias, Berços Dependentes.

Sumário

Agradecimentos	ix
1 Introdução e Objetivos	1
1.1 Introdução	1
1.2 Objetivo	2
1.3 Descrição do Conteúdo dos Capítulos	2
2 Revisão da Literatura	3
2.1 Introdução	3
2.2 Métodos Heurísticos	7
2.3 Métodos Exatos	11
2.3.1 Método de Planos de Corte	11
2.3.2 Método Branch & Bound	12
2.3.3 Método Branch & Cut	12
2.4 Uso de Otimizadores	14
3 Complexo Portuário de Tubarão	15
3.1 Terminal de Praia Mole	17
3.1.1 Dados dos Descarregadores de Navios (DNs) e pás carregadeiras do Píer do Terminal da Praia Mole	18
3.1.2 Dados do Pátio de Estocagem: quantidade e especificações de recuperadoras, empilhadeiras e pás mecânicas	21
3.1.3 Dados das operações da Estação de Carregamento de Vagões	24
3.2 Terminal de Produtos Diversos	25
4 Modelos Matemáticos	27
4.1 Modelagem Matemática para o Terminal de Produtos Diversos	27
4.1.1 Atracação em Berços Dependentes	27
4.1.2 Função Objetivo	30
4.1.3 Formulação Matemática	31
4.2 Modelagem Matemática para o Terminal de Praia Mole	32
4.2.1 Formulação Considerando Navios e Máquinas	33
4.2.2 Tempo de Atendimento	37

4.3	Heurística da Divisão	40
5	Testes e Resultados	42
5.1	Testes	42
5.1.1	Testes para o Terminal de Produtos Diversos	42
5.1.2	Testes para Praia Mole	45
5.2	Resultados	49
5.2.1	Resultados do Terminal de Produtos Diversos	49
5.2.2	Resultados Praia Mole	51
6	Conclusões e Perspectivas	56
	Referências	57
A	Códigos Usados	59
A.1	Terminal de Produtos Diversos	59
A.2	Terminal dePraia Mole	62

Agradecimentos

Meus sinceros agradecimentos a:

Minha família, em especial minha mãe Maria de Fátima Hönigmann e meu pai Paulo Sérgio Cereser, que me apoiaram durante todo meu trajeto até aqui.

Meus orientadores Luiz Leduíno de Salles Neto e Antonio Carlos Moretti que muito me apoiaram no desenvolvimento e elaboração desse projeto.

Ao Instituto de Matemática, Estatística e Computação Científica, e a Fundação de Amparo à Pesquisa do Estado de São Paulo, que financiou esse projeto de mestrado.

A Vale S.A. pelo fornecimento de dados, e pela oportunidade das visitas ao porto de tubarão que foram muito esclarecedoras e produtivas.

Lista de Ilustrações

2.1	Exemplo de Branch & Bound	14
3.1	Porto de Tubarão	15
3.2	Porto de Tubarão: Produtos Diversos e Praia Mole	16
3.3	Esquemática do Terminal de Praia Mole	17
3.4	Terminal de Praia Mole	18
3.5	DNs 1800t/h a esquerda e 2000t/h a direita	19
3.6	Esteira transportadora de Carvão Mineral	19
3.7	Esquemática do Terminal de Praia Mole	21
3.8	Pátios de Estocagem	22
3.9	Empilhadeira e Recuperadora	22
3.10	Formação da Pilha de Carvão Mineral	23
3.11	Formação da Pilha de Coque	24
3.12	Carregamento de Vagões	24
3.13	Terminal de Produtos Diversos	25
3.14	Exemplo de berços dependentes	26
4.1	Situação 2	28
4.2	Situação 3	28

Lista de Tabelas

5.1	Dados do Caso1PD	43
5.2	Dados do Caso2PD	43
5.3	Dados do Caso3PD	44
5.4	Dados do Caso ValePD	45
5.5	Dados do Caso1PM	46
5.6	Dados do Caso2PM	47
5.7	Dados do CasoValePM	48
5.8	Dados do Caso ValePMn	49
5.9	Resultados do Caso1PD	49
5.10	Resultados do Caso2PD	50
5.11	Resultados do Caso3PD	50
5.12	Resultados do Caso Caso1PM	52
5.13	Resultados do Caso2PM	52
5.14	Resultados do Caso ValePM	53
5.15	Resultados do Caso ValePMn	54

Lista de Algoritmos

2.1	Algoritmo com Busca Tabu	9
2.2	Algoritmo com GRASP	10
2.3	Algoritmo PC	11
2.4	Algoritmo B&B	13
4.1	Heurística da Divisão	41

Lista de Códigos

Produtos_Gerais/BercosPG.mos	59
Praia_Mole_Fila_Livre/BercosPM.mos	62

Lista de Símbolos

M Representa um número muito grande.

t Representa Toneladas.

h Representa Hora.

t/h Representa Toneladas por Hora.

Lista de Abreviaturas e Siglas

PAB Problema de Alocação de Navios em Berços.

PABP Problema de Alocação de Navios em Berços com Prioridades.

PIB Produto Interno Bruto.

PAG Problema de Alocação de Guindastes.

PQA Problema Quadrático de Alocação.

FIFO *First In First Out*.

DN Descarregador de Navio.

TRTC ou TC Transportadores de Carvão ou Esteiras.

EP Empilhadeira.

RC Recuperadora.

ECV Estação de Carregamento de Vagões.

TPM Terminal de Praia Mole.

TPD Terminal de Produtos Diversos.

CPLEX Pacote Computacional de Otimização (*IBM ILOG CPLEX Optimization Studio*).

Capítulo 1

Introdução e Objetivos

Nesse capítulo estão a introdução e os objetivos desse projeto de pesquisa.

1.1 Introdução

No Brasil o setor portuário é responsável por 95% do volume do comércio exterior. Se mantidas as taxas de crescimento de 5% dos últimos anos, os portos brasileiros terão que aumentar sua capacidade de atendimento de 650 milhões de toneladas por ano, em 2012, para 900 milhões em 2017. Esse acréscimo de 40%, ou 32 milhões de toneladas por ano, equivale a um porto de Santos a cada três anos.

Os custos logísticos com transporte de cargas representam 17,3% do PIB no Brasil, enquanto nos Estados Unidos, encontra-se em 8% e, na Ásia, em 9,5%. Os custos portuários no Brasil chegam a ser até quatro vezes maiores que os dos portos asiáticos e europeus.

A navegação de cabotagem brasileira transportou 138 milhões de toneladas em mercadorias em 2012, número 4% superior à quantidade transportada em 2011, quando foram transportados 133 milhões de toneladas, segundo o Anuário Estatístico Aquaviário. Do total transportado na cabotagem brasileira no ano passado, 79% corresponderam a cargas de granel líquido, com 109 milhões de toneladas; 12% a granel sólido (17 milhões de toneladas); 5% a carga geral containerizada (7 milhões de toneladas); e 3,5% a carga geral solta (4,8 milhões de toneladas).

Em 2012, os gastos do País com afretamentos de embarcações somaram cerca de US\$ 6 bilhões, representando um acréscimo de 18,8% em relação a 2011. No apoio portuário, os gastos com afretamento somaram US\$ 49,1 milhões no ano passado; no apoio marítimo, US\$ 3,09 bilhões; na cabotagem, US\$ 114,4 milhões; e no longo curso, US\$ 2,7 bilhões, registrando um crescimento em termos de gastos de 2,1%, 22,8%, 9,5% e 16,2% em comparação com o ano anterior, respectivamente. Notamos que, em quase todos os casos, os custos aumentaram percentualmente muito mais que o volume percentual transportado.

Tendo em vista esses valores, fica claro que o Brasil necessita de um grande aumento da capacidade portuária, juntamente com uma grande redução dos custos portuários. Se aplicar o crescimento necessário de 40%, com a atual política de gerenciamento dos portos, além do alto custo de ampliação da infraestrutura portuária, isso acarretaria em um aumento de 188% nos custos

operacionais portuários, valor que pode ser considerado impraticável tendo em vista a previsão de investimento na área.

Nesse contexto torna-se necessário o estudo de uma melhor política portuária, a fim de atingir as metas de crescimento do volume de comércio, e de redução dos já altos custos portuários Brasileiros.

1.2 Objetivo

Visamos com o desenvolvimento deste projeto:

- desenvolver novos modelos e métodos computacionais para o problema de alocação de navios em berços para carregamento/descarregamento, integrado com o problema de alocação de navios e máquinas portuárias a berços;
- desenvolver novos métodos para obtenção de soluções eficientes para o problema integrado de carregamento/descarregamento de navios com múltiplos objetivos;

1.3 Descrição do Conteúdo dos Capítulos

- No capítulo 2 dessa dissertação denominado "Revisão da Literatura" contém o resumo de alguns dos trabalhos feitos na área de alocação de berços.
- No capítulo 3 denominado "Complexo Portuário de Tubarão" contém a descrição de características, processos, equipamentos dentre outros, principalmente do terminal de Praia Mole localizado nesse complexo portuário.
- No capítulo 4 denominado "Modelos Matemáticos" estão descritos os modelos matemáticos desenvolvidos nesse trabalho e também uma heurística para a solução dos modelos. Esses modelos e heurística foram desenvolvidos e criados nesse projeto de pesquisa e esse modelos, bem como todo o trabalho em si, estão em processo de patente por parte da empresa Vale S.A..
- No capítulo 5 denominado "Testes e Resultados" estão descritos os testes e também os resultados obtidos, desde testes e resultados triviais até testes e resultados de casos reais, baseados em dados fornecidos pela empresa Vale S.A..
- No capítulo 6 denominado "Conclusões e Perspectivas" contém uma análise dos modelos, resultados obtidos e da heurística desenvolvidos nesse trabalho.

Capítulo 2

Revisão da Literatura

Nesse Capítulo são apresentadas algumas das revisões Bibliográficas feitas para o desenvolvimento desse projeto.

2.1 Introdução

Em muitos portos é importante estabelecer qual ordem e como os navios serão descarregados ou carregados. O problema de alocação de navios em berços possui duas decisões principais *onde* e *quando* os navios devem atracar. Normalmente, existem restrições de atracação (devido a profundidade de cada berço), também com relação a distância máxima entre guindaste e navio, ou até mesmo se o berço é especializado em um tipo de navio específico. Essa escolha visa minimizar o tempo de espera dos navios no porto.

Definindo:

- N : conjunto de navios, $n = |N|$;
- M : conjunto de berços, $m = |M|$;
- t_i^k : duração do atendimento do navio i no berço k ;
- a_i : horário de chegada do navio i ;
- s^k : horário de abertura do berço k ;
- e^k : horário de fechamento do berço k ;
- b_i : horário de termino da janela de tempo para o navio i ;
- v_i : valor do tempo de serviço do navio i ;
- $x_{ij}^k \in \{0, 1\} \forall k \in M, \forall (i, j) \in A^k, x_{ij}^k = 1$ se o navio j é atendido pelo berço k após o navio i ;
- $T_i^k \forall k \in M, i \in N$ o horário que o navio i atracou no berço k ;

- $T_{o(k)}^k \forall k \in M$ o horário em que o primeiro navio atracou no berço k ;
- $T_{d(k)}^k \forall k \in M$ o horário em que o último navio saiu do berço k ;
- $M_{ij}^k = \max\{b_i + t_i^k - a_i\}, \forall k \in M, \forall (i, j) \in N$.

Segundo Cordeau et al. [1] podemos descrever o problema de alocação de navios em berços matematicamente da seguinte forma: a função objetivo (2.1.1a) minimiza a soma dos tempos de espera; a restrição (2.1.1b) garante que cada navio é atendido por apenas um berço; as restrições (2.1.1c) e (2.1.1d) garantem, respectivamente, que um navio será o primeiro a ser atendido em cada berço e outro será o último; a restrição (2.1.1e) garante a conservação do fluxo, ou seja, o atendimento para os demais navios; a restrição (2.1.1f) faz o cálculo do horário de atracação dos navios, sendo considerados apenas os navios válidos para cada berço, ou seja, alguns navios não podem ser atendidos em determinados berços devido a restrições técnicas; as restrições (2.1.1g) e (2.1.1h) garantem, respectivamente, que o horário de atracação seja após a chegada do navio e que o horário do término do atendimento do navio seja anterior ao horário limite do navio (janela de tempo); as restrições (2.1.1i) e (2.1.1j) garantem a não violação das janelas de tempo nos berços; por fim, a restrição (2.1.1k) garante que as variáveis de decisão sejam binárias.

$$\min \sum_{i \in N} \sum_{k \in M} v_i \left[T_i^k - a_i + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{i,j}^k \right] \quad (2.1.1a)$$

$$\text{s.a.} \sum_{k \in M} \sum_{j \in N \cup \{d(k)\}} x_{ij}^k = 1 \quad \forall i \in N \quad (2.1.1b)$$

$$\sum_{j \in N \cup \{d(k)\}} x_{o(k),j}^k = 1 \quad \forall k \in M \quad (2.1.1c)$$

$$\sum_{i \in N \cup \{o(k)\}} x_{i,d(k)}^k = 1 \quad \forall k \in M \quad (2.1.1d)$$

$$\sum_{j \in N \cup \{u(k)\}} x_{i,j}^k - \sum_{j \in N \cup \{o(k)\}} x_{j,i}^k = 0 \quad \forall k \in M, \forall i \in N \quad (2.1.1e)$$

$$T_i^k + t_i^k - T_j^k \leq (1 - x_{i,j}^k) M_{i,j}^k \quad \forall k \in M, \forall (i, j) \in A^k \quad (2.1.1f)$$

$$T^k \geq a_i \quad \forall k \in M, \forall i \in N \quad (2.1.1g)$$

$$T_i^k + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{j,i}^k \leq b_i \quad \forall k \in M, \forall i \in N \quad (2.1.1h)$$

$$T_{o(k)}^k \geq s^k \quad \forall k \in M \quad (2.1.1i)$$

$$T_{d(k)}^k \leq e^k \quad \forall k \in M \quad (2.1.1j)$$

$$x_{i,j}^k \in \{0, 1\} \quad \forall k \in M, \forall (i, j) \in A^k \quad (2.1.1k)$$

A função objetivo minimiza a soma dos tempos de atendimento de acordo com um peso v_i que representa a prioridade de cada navio.

Em Imai et al. [9] é desenvolvido o Problema de alocação de berços (PAB) de forma dinâmica. São consideradas duas formulações, uma sem prioridades para os navios, e outra com essa prioridade. A formulação sem a atribuição de prioridades é feita para estudar as dificuldades da formulação e da metodologia de resolução.

Existem alguns critérios para a medida de produtividade do berço, como tempo de serviço e tempo de espera do navio por um berço. A formulação está muito mais relacionada com as operações do porto do que com seus usuários. O PAB assume um longo cais em um terminal de multiúso, considera a variedade de navios de diferentes tamanhos, comprimentos, e grande número de alternativas de atracação ao longo do cais. Para simplificar, o cais é dividido em diversos blocos, e no PAB obtemos um conjunto de alocação de navios nesses blocos chamados de berços.

Assumimos que cada berço pode atender um navio de cada vez, independente das características desse navio e também não foi considerado que não havia nenhuma restrição para a atracação dos navios ao longo do cais. Foi considerado que o tempo de serviço dependia do berço em que o navio atraca. Também foi considerado que as cargas podem chegar ao berço após a atracação do navio que vai ser carregado com elas. Também é considerada a distância entre o berço e o depósito onde está a carga do navio. Além disso o modelo garante o atendimento consecutivo para todos os navios sem interrupções.

O PAB pode ser formulado como um problema inteiro misto, dado por:

$$\min \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} \{(T - k + 1)C_{i,j} + S_i - A_j\} x_{i,j,k} + \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} (T - k + 1)y_{i,j,k} \quad (2.1.2a)$$

$$\text{s.a} \sum_{i \in B} \sum_{k \in U} x_{i,j,k} = 1 \quad \forall j \in V \quad (2.1.2b)$$

$$\sum_{j \in V} x_{i,j,k} \leq 1 \quad \forall i \in B, k \in U \quad (2.1.2c)$$

$$\sum_{l \in V} \sum_{m \in P_k} (C_{i,l}x_{i,l,m} + y_{i,l,m}) + y_{i,j,k} - (A_j - S_i)x_{i,j,k} \geq 0 \quad \forall i \in B, j \in W_i, k \in U \quad (2.1.2d)$$

$$x_{i,j,k} \in \{0, 1\} \quad \forall i \in B, j \in V, k \in U \quad (2.1.2e)$$

$$y_{i,j,k} \geq 0 \quad \forall i \in B, j \in V, k \in U \quad (2.1.2f)$$

onde $i \in B$ é o conjunto dos berços, $j \in V$ é o conjunto dos navios, $k \in U$ é o conjunto da sequência de serviço dos navios, A_j é a hora de chegada do navio j , P_k é o subconjunto de U tal que $P_k = \{p | p < k \in U\}$, S_i é o tempo de abertura do berço i , W_i é o subconjunto dos navios tais que $A_j \geq S_i$, $C_{i,j}$ é o tempo que o navio j gasta no berço i , $x_{i,j,k} = 1$ se o navio j é o k -ésimo atendido pelo berço i e $x_{i,j,k} = 0$ caso contrário, $y_{i,j,k}$ é o tempo que o berço i ficou ocioso entre o atendimento do $(k - 1)$ -ésimo navio e o k -ésimo navio desse berço.

Objetivo (2.1.2a) minimiza a soma dos tempos de espera e o serviço para cada navio. Conjunto de restrições (2.1.2b) garante que cada navio deve ser atendido em algum berço em qualquer ordem de serviço. As restrições (2.1.2c) garante que todo navio é atendido em algum momento. As restrições (2.1.2d) asseguram que os navios são atendidos após a sua chegada.

Para a formulação com prioridade de atendimento Imai et al. [9] argumentam que os navios com diferentes cargas, tamanhos, volumes, etc, devem possuir diferentes prioridades para o atendimento.

O tempo de espera de cada navio, é o tempo de manobra cumulativo de todos os navios anteriores ao mesmo berço. Isto é, se o serviço do navio 1 é seguido por 2 e 3, o tempo de manobra do navio 1 é uma parte do tempo de serviço para os navios de 2 e 3. A consideração de prioridade é tido por um parâmetro de ponderação para o atendimento de cada navio. Assim, o tempo de manobra do navio 1 contribui para tempos de serviço seu sucessor ponderado pelo peso atribuído.

$$\min \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} \left\{ C_{i,j} + S_i - A_j + \sum_{l \in V} \sum_{m \in P_k} C_{i,l} x_{i,l,m} \right\} \alpha_j x_{i,j,k} \quad (2.1.3a)$$

$$+ \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} \left\{ y_{i,j,k} + \sum_{l \in V} \sum_{m \in P_k} y_{i,l,m} \right\} \alpha_j \quad (2.1.3b)$$

$$\text{s.a} \sum_{i \in B} \sum_{k \in U} x_{i,j,k} = 1 \quad \forall j \in V \quad (2.1.3c)$$

$$\sum_{j \in V} x_{i,j,k} \leq 1 \quad \forall i \in B, k \in U \quad (2.1.3d)$$

$$\sum_{l \in V} \sum_{m \in P_k} (C_{i,l} x_{i,l,m} + y_{i,l,m}) + y_{i,j,k} - (A_j - S_i) x_{i,j,k} \geq 0 \quad \forall i \in B, j \in W_i, k \in U \quad (2.1.3e)$$

$$x_{i,j,k} \in \{0, 1\} \quad \forall i \in B, j \in V, k \in U \quad (2.1.3f)$$

$$y_{i,j,k} \geq 0 \quad \forall i \in B, j \in V, k \in U \quad (2.1.3g)$$

onde α_j é a prioridade de atendimento do navio j .

Para a resolução do PABP, Imai et al. [9] fazem uso da relaxação Lagrangiana, colocando a restrição (2.1.3e) na função objetivo e elimina a variável $y_{i,j,k}$ já que ela não aparece em nenhuma restrição, sendo redundante para o problema, assim a função objetivo fica:

$$\min \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} \left\{ C_{i,j} + S_i - A_j + \sum_{l \in V} \sum_{m \in P_k} C_{i,l} x_{i,l,m} \right\} \alpha_j x_{i,j,k} \quad (2.1.4a)$$

$$+ \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} \lambda_{i,j,k} (A_j - S_i) x_{i,j,k} - \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} \sum_{l \in V} \sum_{m \in P_k} \lambda_{i,j,k} C_{i,l} x_{i,l,m} \quad (2.1.4b)$$

Definindo $D_{i,j,k}^* = (C_{i,j} + S_i - A_j)\alpha_j + \lambda_{i,j,k}(A_j - S_i)$, $D_{i,j,l}^\# = \lambda_{i,j,k}C_{i,l}$ e $E_{i,j,l} = \alpha_j C_{i,l}$ a função objetivo é reformulada como:

$$\min \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} \left\{ D_{i,j,k}^* + \sum_{l \in V} \sum_{m \in P_k} E_{i,j,l} x_{i,l,m} \right\} x_{i,j,k} - \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} \sum_{l \in V} \sum_{m \in P_k} D_{i,j,k}^\# x_{i,l,m} \quad (2.1.5a)$$

$$= \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} D_{i,j,k}^* x_{i,j,k} - \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} \sum_{l \in V} \sum_{m \in P_k} D_{i,j,k}^\# x_{i,l,m} + \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} \sum_{l \in V} \sum_{m \in P_k} E_{i,j,k} x_{i,l,m} x_{i,j,k} \quad (2.1.5b)$$

Como em Imai et al. [8] a formulação se torna bidimensional :

$$\min \sum_{j,n} D'_{j,n} x_{j,n} + \sum_{j,n} \sum_{p,q} E'_{p,q,j,n} x_{p,q} x_{j,n} \quad (2.1.6a)$$

$$\text{s.a } \sum_n x_{j,n} = 1 \quad \forall j \quad (2.1.6b)$$

$$\sum_j x_{j,n} \leq 1 \quad \forall n \quad (2.1.6c)$$

$$x_{j,n} \in \{0, 1\} \quad \forall j, n \quad (2.1.6d)$$

onde j e p são índices de navios e n e q são recursos (isto é berços e serviço). (2.1.6) é um problema de quadrático (PQA) para o qual é difícil encontrar a solução ótima, pois é um problema NP-difícil. Devido a esse fato, o PQA tem sido resolvido de forma aproximada.

Outra forma de alocação de berço consiste em particionar o espaço do berço e/ou do porto e o planejamento é feito sobre cada espaço particionado, dessa forma um navio atracado pode estar sendo atendido e outro navio também atracado espera para ser atendido. Nessa forma de alocação o primeiro objetivo tratado foi minimizar o máximo espaço usado.

Na próxima seção abordamos alguns métodos heurísticos para resolução do PAB. Na Seção 3 apresentamos alguns métodos utilizados para resolução desse problema.

2.2 Métodos Heurísticos

Para pequenos casos as formulações dessa seção podem ser resolvidas pelo CPLEX. No entanto, nenhum dos modelos possui tempo de solução aceitável para casos reais. Devido a esse fato, diversos trabalhos buscam heurísticas eficientes para a resolução das diversas formas do PAB.

Cordeau et al. [1] desenvolve uma heurística baseada na busca tabu para resolver o problema do PAB discreto. Denote por S o conjunto de soluções que satisfazem as restrições do PAB (2.1.1b) - (2.1.1g) e (2.1.1i). A heurística explora o espaço das soluções, movendo a cada iteração a partir da solução atual s para a melhor solução na sua vizinhança da atual solução s denotada $N(s)$. Cada solução $s \in S$ é representada por um conjunto de m sequências de berços tais que cada navio pertence exatamente a uma sequência. No entanto, esta solução pode violar as restrições de janela de tempo associadas aos navios e aos berços. A restrição de janela de tempo do navio i no berço k é violada se o tempo de conclusão $T_{i,k} + t_{i,k}$ do navio i é maior do que o limite superior da janela de tempo b_i . A atracação antes de a_i não é permitida, em outras palavras, $T_{i,k} \geq a_i$. Da mesma forma, a janela de tempo do berço k é violada quando o tempo de conclusão de um navio i atribuído a atracar no berço k é maior do que o limite superior da janela de tempo do cais (fechamento do berço). Em seguida, as soluções são avaliadas por meio de uma função objetivo penalizada de acordo com o grau da violação das restrições.

Para cada solução $s \in S$ é associado um conjunto $B(s) = \{f(i, k): \text{ o navio } i \text{ atribuído ao berço } k\}$. A vizinhança $N(s)$ de uma solução s é definida pela aplicação de um simples operador que remove um atributo (i, k) de $B(s)$ e substitui-lo por outro atributo (i, k') , onde $k \neq k'$. A

solução inicial é construída relaxando as restrições de janela de tempo. Estas soluções iniciais são normalmente inviáveis, mas isto não é um problema para a busca heurística tabu.

A meta heurística conhecida como Busca Tabu é um procedimento adaptativo auxiliar, que guia um algoritmo de busca local na exploração contínua dentro de um espaço de busca. Diferente do que pode vir a ocorrer com outros procedimentos, a Busca Tabu não é confundida pela ausência de vizinhos aprimorantes. Isso significa que o método evita retornar a um ótimo local visitado previamente, de forma a superar a otimalidade local e atingir um resultado ótimo ou próximo ao ótimo global.

Partindo de uma solução inicial, que pode ser escolhida de acordo com algum dentre os vários critérios possíveis, a busca move-se, a cada iteração, para a melhor solução na vizinhança, não aceitando movimentos que levem a soluções já visitadas por permanecerem armazenadas em uma lista tabu. A lista permanece na memória guardando soluções já visitadas (tabu) durante um determinado espaço de tempo ou certo número de iterações (prazo tabu). Como resultado final é esperado que se encontre um ótimo global ou a solução mais próxima deste.

A Busca Tabu utiliza estruturas de memória flexíveis para armazenar o conhecimento referente aos espaços que já foram percorridos. Devido ao fato de utilizar intensivamente técnicas de memória adaptativa, a técnica consegue proporcionar boas soluções, onde boas denota soluções geralmente ótimas ou perto do ótimo global em um tempo de processamento aceitável.

Outra importante característica do método é que a solução final tem pouca ou nenhuma dependência da escolha feita para a solução inicial. Isso graças aos mecanismos implementados pelo método, que fogem de ótimos locais

Voltando ao problema, a busca começa a partir da solução produzida pelo procedimento de obter uma solução inicial e seleciona, em cada iteração, a melhor solução não tabu $s \in N(s)$. Depois de cada iteração, o valor do parâmetro α é modificado por um fator $1 + \tau$, onde $\tau > 0$. Se a solução atual é possível no que diz respeito às restrições de janela de tempo, o valor de α é dividido por $1 + \tau$, caso contrário, é multiplicado por $1 + \tau$. Este processo é repetido para n iterações e a melhor solução viável s^* é escolhida ao longo da busca. O algoritmo reinicia com a solução fornecida pelo procedimento FCFS-G. O algoritmo completo pode ser resumido como:

Algoritmo 2.1 Algoritmo com Busca Tabu

faça $h \leftarrow 1$

Fase de inicialização: obter uma solução $s \in S$ utilizando o procedimento de RG, se $h = 1$, caso contrário, use o procedimento FCFS-G.

Definir $\alpha = 1$. Se s for viável, faça $s^* = s$ e $c(s^*) = c(s)$, caso contrário, faça $c(s^*) = \inf$

for $j := 1, \dots, n$ **do**

(a) Seleção de uma solução $\hat{s} \in N(s)$ que minimiza $f(s) + p(s)$ e não é tabu ou satisfaz seus critérios de aspiração.

(b) Se a solução s é viável e $c(s) < C(S^*)$; então $s^* = s$ e $c(s^*) = c(\hat{s})$

(c) Calcular $w(s)$ e atualizar α conforme.

(d) Definir $s = \hat{s}$.

end for

if $h = 1$ **then**

$h = 2, s_1^* = s^*$

Va para Fase de inicialização

end if

Faça $s^* = \min\{s_1^*, s^*\}$ e pare.

Em Lopes et al. [12], o PAB foi tratado por meio do modelo relaxado. Para sua resolução foi usado o GRASP+PR proposto por Feo and Resende [4] e Glover [7], respectivamente.

Proposta por Feo and Resende [4], a meta-heurística GRASP *Greedy Randomized Adaptive Search Procedure* é um algoritmo iterativo onde cada iteração é composta por duas fases: uma fase de construção onde uma solução viável para o problema é criada e uma fase de busca local, posterior à construção, que tem como objetivo tentar melhorar a solução inicial obtida na fase de construção. As iterações GRASP são independentes, ou seja, na iteração atual não se leva em conta nenhuma informação das iterações anteriores. O critério de parada normalmente usado é um número máximo de iterações. A melhor solução obtida ao final da execução do GRASP é a solução final.

Path Relinking (PR) foi originalmente proposto no contexto da Pesquisa Tabu meta-heurístico Glover [7], em que um dos principais objetivos é o de fazer um balanço entre a intensificação e busca diversificação. PR é baseado na geração de novas soluções, explorando trajetórias que conectam soluções de elevada qualidade, a partir da pesquisa de uma dessas soluções, denominada solução inicial e gerando um caminho na vizinhança espacial que dirige a pesquisa para outras soluções, as soluções de chamada guia.

Para a resolução do problema de alocação de navios em berços, Lopes et al. [12] através do método GRASP encontrada uma solução viável e em seguida um ótimo local por meio de buscas. Após isso o PR *Path Relinking* é aplicado, explorando as trajetórias que conectam a solução da iteração atual é melhor solução obtida até então. A cada iteração do GRASP, é gerada uma solução por meio de uma heurística de construção, na qual é aplicada uma busca local e, a cada intervalo de iterações, é aplicado o PR.

A heurística de construção cria uma lista N de navios por ordem de chegada e uma lista B com todos os berços. A partir destas listas, os navios da lista N são atribuídos a um berço selecionado de forma aleatória, mas que seja capaz de atender ao navio em questão.

Após a criação de uma solução s , proveniente da fase de construção, é aplicada uma busca local e, em cada uma de suas iterações, é aplicado um movimento de troca em s gerando uma solução vizinha s' . Os movimentos utilizados foram os mesmos propostos por [?]: Reordenar Navios, Realocar Navios e Trocar Navios. Assim, após várias iterações, a melhor solução é retornada.

O movimento Reordenar Navios faz uma simples troca na sequência de atendimento de um berço, selecionado aleatoriamente. Ou seja, para um berço qualquer, dois navios são selecionados e suas posições na sequência de atendimento são trocadas.

Já o movimento de *Realocar Navio* consiste em selecionar dois berços quaisquer, pertencentes à solução s , selecionar um navio qualquer em um dos berços, extraí-lo de seu berço atual e atribuí-lo ao outro berço; caso o berço onde o navio será atribuído não possa atendê-lo, outro berço deverá ser selecionado. Após a atribuição, a lista de atendimento do novo berço deverá ser ordenada pelo horário de chegada dos navios.

Por fim, o movimento *Trocar Navios* consiste em selecionar dois berços quaisquer, pertencentes à solução s , selecionar um navio qualquer em cada um dos berços e trocá-los. Assim como no movimento de Realocar navio, caso os navios não possam ser atendidos pelos berços onde serão alocados, deverão ser selecionados outros navios e/ou berços. Após a troca, a lista de atendimento dos dois berços deverão ser ordenada pelo horário de chegada dos navios.

O PR funciona como uma estratégia para explorar as trajetórias que conectam a solução corrente com a melhor solução encontrada. Assim, realizadas $iterPR$ iterações, a solução corrente s é progressivamente transformada na melhor solução s^* , ou seja, para cada navio, se os seus berços, em ambas as soluções forem diferentes, então o navio da solução corrente s será remanejado para o mesmo berço (e posição de atendimento) em que é atendido na melhor solução s^* .

O algoritmo do método que faz uso do GRASP+PR aqui descrito está resumido abaixo;

Algoritmo 2.2 Algoritmo com GRASP

```

 $f^* \leftarrow \text{inf}$ 
 $j \leftarrow 0$ 
for  $i \leftarrow 1$  to maxit do
   $s \leftarrow \text{HEURISTICA\_DE\_CONSTRUCAO}$ 
   $s \leftarrow \text{BUSCA\_LOCAL}(s, \text{tamBL})$ 
  if  $f(s) < f^*$  then
     $f^* \leftarrow f(s)$ 
     $s^* \leftarrow s$ 
  end if
  if ( $j = \text{iterPR}$ ) then
     $s^* \leftarrow \text{PR}(s, s^*)$ 
     $f^* \leftarrow f(s^*)$ 
     $j \leftarrow 0$ 
  end if
end for
return ( $s^*$ )

```

2.3 Métodos Exatos

O problema de alocação de navios em Berços é Problema linear inteiro misto (Jarvis and Serali [10] e Luenberger:2003). Existem dois métodos gerais para encontrar a solução exata de um problema linear inteiro misto :

1. Método de planos de corte *cutting plane* - PC
2. Método tipo *branch and bound*- B&B
3. Metodo de *Branch an Cut*

Muitos códigos computacionais comerciais usam o B&B, alguns aplicam o PC como ajuda adicional.

2.3.1 Método de Planos de Corte

O método de planos de corte é um método exato para Programação linear que busca iterativamente refinar um conjunto viável ou função objetivo por meio de inequações lineares, chamadas cortes. Tais procedimentos são utilizados para encontrar soluções inteiras para problemas de Programação Linear Inteira (PLI), bem como para resolver problemas gerais de otimização convexa. O uso de planos de corte para resolver PLI foi introduzido por Ralph Gomory Gilmore and Gomory [6].

Tenso sempre em vista o problema linear inteiro:

$$Maxz = c'x$$

$$sujeito a : Ax = b, x \text{ inteiro não - negativo}$$

Um algoritmo do tipo Planos de Corte:

Algoritmo 2.3 Algoritmo PC

Inicialização: Resolver o PLI relaxado - PLIR (sem as restrições de integralidade)

Fazer o Teste de otimalidade:

if Solução ótima do PLIR for inteira **then**

 A solução ótima do PLI foi encontrada - Parar

else

 Acrescentar uma nova restrição (corte) ao PLIR

 Voltar para inicialização

end if

Um método que faz uso do plano de cortes é o Corte Fracionário de Golmory para PLI puro Gilmore and Gomory [6] e Nemhauser and Wolsey [15], dado por:

$$- \sum_{j \in N} f_{k,j} X_j \leq -f_k$$

onde

- f_k - Parte fracionária de b_k .
- $f_{k,j}$ - Parte fracionária de a_{kj} .
- N - Conjunto das variáveis não-básicas.

2.3.2 Método Branch & Bound

O princípio do método *Branch-and-Bound* (BB) Mitchell [13] é uma enumeração de todas as soluções viáveis de um problema de otimização combinatorial de minimização ou maximização, tal que propriedades ou atributos não compartilhados por qualquer solução ótima são detectados previamente. Um atributo, chamado também de ramo da árvore de enumeração, define um subconjunto do conjunto de todas as soluções viáveis do problema original onde cada elemento do subconjunto satisfaz este atributo. Apenas uma fração das soluções factíveis é realmente examinada, visto que, conforme afirma Render et al. [16], o método *Branch-and-Bound* é um algoritmo que busca por uma solução ótima através do exame de somente uma pequena parte do número total de possíveis soluções, pois ele trabalha quebrando o espaço de soluções viáveis em subproblemas menores até que uma solução ótima seja alcançada.

O termo *branch* refere-se ao fato de que o método efetua partições no espaço das soluções, ou seja, divide um problema em problemas menores. Uma maneira de se criar estes subproblemas usando fixação de variáveis discretas.

2.3.3 Método Branch & Cut

O Método *Branch & Cut* é um procedimento eficiente de enumeração das possíveis soluções viáveis inteiras proposto em Land and Doig [11], tem por base os métodos *Branch-and-Bound* e *Cutting Plane*. Seja o PLI :

$$\text{Max} z = c'x$$

$$\text{sujeito a : } Ax = b, x \text{ inteiro não - negativo}$$

No problema, x_j inteiro para $j \in I$, onde I = conjunto das variáveis inteiras.

Abaixo se encontra um algoritmo do tipo *Branch & Cut*:

Algoritmo 2.4 Algoritmo B&B

Resolva o PLI sem as restrições de integralidade. Denote por PL-1 com solução ótima = z_1

if Todo x_j inteiro para $j \in I$ **then**

solução ótima para o PLI

Parar!

else

Existe x_j fracionário para $j \in I$, e z_1 é um limitante superior *upper bound* para o valor da função objetivo ótima do PLI.

Particionar (*branch*) a região viável do PL-1 com respeito a algum x_j fracionário para $j \in I$, procurando achar soluções inteiras que serão limitantes inferiores (*lower bound*) para o z do PLI.

Suponha que x_j foi selecionada para a partição e que β_j é o seu valor (é um valor fracionário nesse caso).

Criar dois novos problemas, PL-2 e PL-3, introduzindo as restrições $x_j \leq \lfloor \beta_j$ e $x_j \geq \lceil \beta_j$, onde $\lfloor \beta_j$ é o maior inteiro, menor que β_j e $\lceil \beta_j$ é o menor inteiro maior que β_j

PL-2

$$\text{Max } z = c'x$$

$$\text{sujeito a : } Ax = b, x \geq 0, x_j \leq \lfloor \beta_j$$

PL-3

$$\text{Max } z = c'x$$

$$\text{sujeito a : } Ax = b, x \geq 0, x_j \geq \lceil \beta_j$$

Usar o Método Dual Simplex Jarvis and Sherali [10] para encontrar as soluções ótimas dos PL-2 e PL-3.

if Estas soluções ainda são fracionárias **then**

Selecionar PL-2 ou PL-3 para desenvolver uma nova ramificação com a introdução de uma nova restrição, como foi feito anteriormente.

if Solução inteira para algum dos PLs criados **then**

O valor ótimo da função objetivo desta solução inteira é um limitante inferior (*lower bound*) para o valor máximo de z para o PLI original.

end if

Admitir como explorados (*fathomed*) aqueles PLs cujos valores de z não são melhores que o *lower bound* conhecido, ou que sejam inviáveis, ou ainda que já tenham solução ótima inteira.

end if

end if

Para melhor ilustrar o funcionamento do *Branch & Bound*

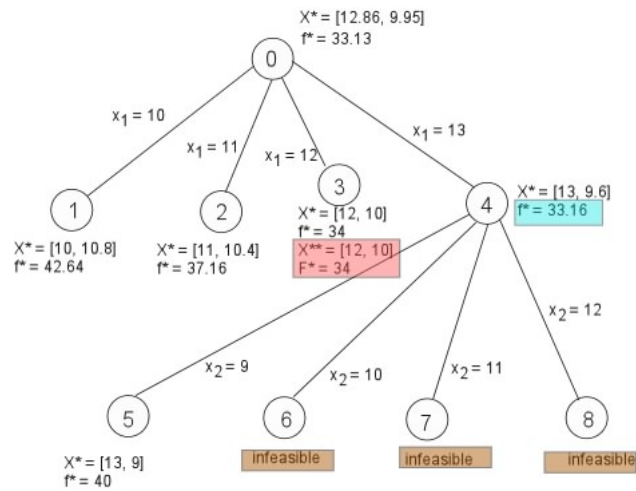


Figura 2.1: Exemplo de Branch & Bound

2.4 Uso de Otimizadores

O pacote de otimização CPLEX [2] é uma das melhores ferramentas para resolução de problemas lineares inteiros da atualidade, é baseado no método Simplex Jarvis and Serali [10] e faz uso da técnica de *branch-and-bound* (busca em árvore), *cutting plane* e a mistura dos dois *branch & cut* para encontrar a solução ótima inteira de problema do tipo lineares inteiros.

Com o objetivo de validar o modelos aqui propostos, faremos uso da ferramenta XpressTM Xpress [17] que possui um algoritmo tipo simplex pré-programado para a solução de problemas lineares inteiros.

Capítulo 3

Complexo Portuário de Tubarão

Nesse capítulo se encontram as características do complexo portuário de tubarão e seus terminais.



Figura 3.1: Porto de Tubarão

O complexo portuário de Tubarão localizado na ponta do mesmo nome, na parte continental do município de Vitória, capital do Espírito Santo. Inaugurado em 1966, é controlado pela Companhia Vale do Rio Doce. é o maior porto de exportação de minério de ferro do mundo e permite o acesso de navios Graneleiros de grande porte. Possui 4 terminais de carga/descarga:

- TPM - Terminal de Praia Mole (carvão)
- TPD - Terminal de Produtos Diversos (grãos e fertilizantes)
- TGL - Terminal de Granéis Líquidos (combustíveis)
- Terminal de Tubarão (minério e pelotas)

Nesse capítulo também está descrito o terminal de Praia Mole e uma breve descrição do terminal de Produtos Diversos.



Figura 3.2: Porto de Tubarão: Produtos Diversos e Praia Mole

3.1 Terminal de Praia Mole

No Terminal de Praia Mole, só é realizado o descarregamento de navios do tipo graneleiro, com carga de carvão mineral. O terminal de Praia Mole opera com guindastes do tipo Descarregador de Navio, que se movimenta em um trilho, possui 2 berços e faz o transporte do carvão do navio para o pátio de estocagem e do pátio do estocagem até a estação de carregamento de trens, através de esteiras, chamadas de Linhas. A figura 3.3 é uma esquemática do termina de Praia Mole

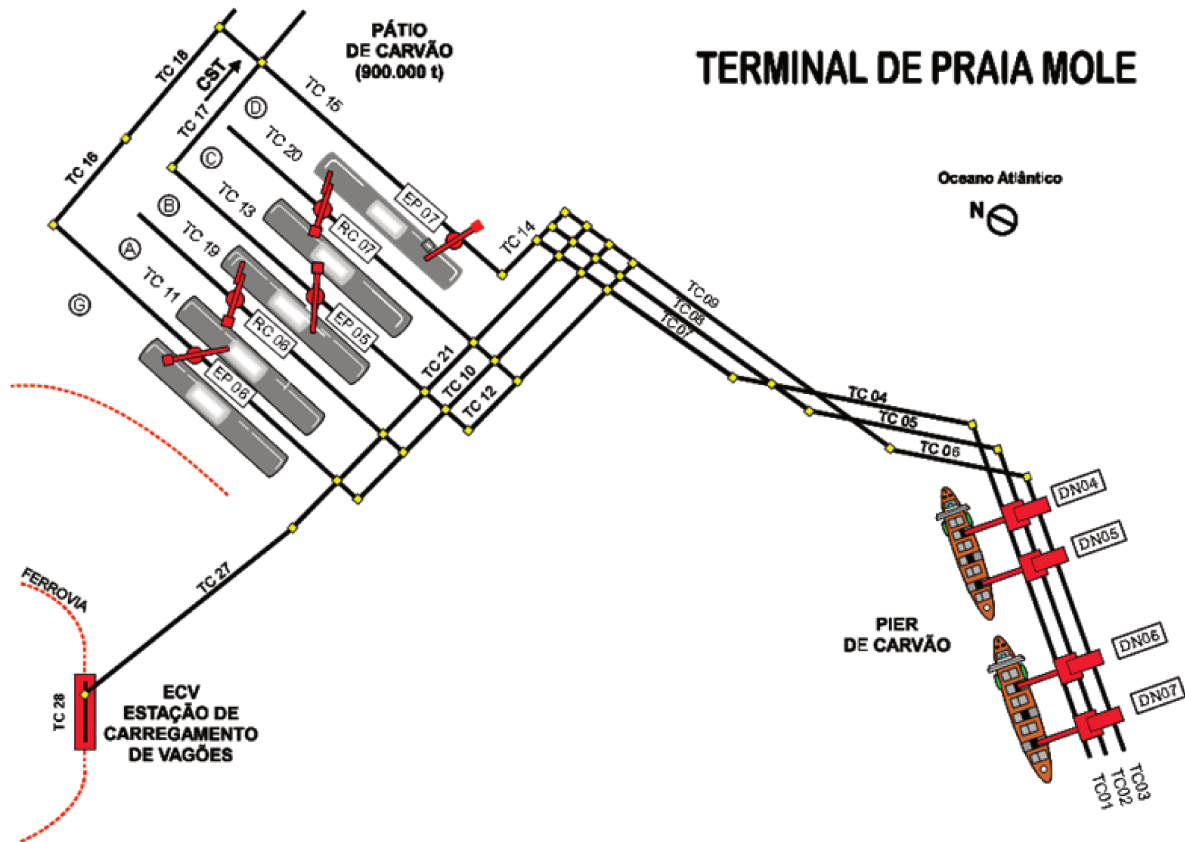


Figura 3.3: Esquemática do Terminal de Praia Mole

A a figura 3.4 uma foto real do termina de Praia Mole



Figura 3.4: Terminal de Praia Mole

3.1.1 Dados dos Descarregadores de Navios (DNs) e pás carregadeiras do Píer do Terminal da Praia Mole

- O Terminal de Praia Mole contém 4 DNs, denominados DN04, DN05, DN06 e DN07 respectivamente, da esquerda para a direita na figura 3.4.
 - Os DNs04 e 05 possuem capacidade de desembarque de 1800 t/h;
 - Os DNs06 e 07 possuem capacidade de desembarque de 2000 t/h;



Figura 3.5: DNs 1800t/h a esquerda e 2000t/h a direita

- Existem 4 DNs e apenas 3 linhas para o desembarque. Cada DN permite levar carvão mineral para o pátio e pode operar em qualquer uma das três esteiras disponíveis para o transporte. É possível a operação de 2 DNs em uma mesma linha.



Figura 3.6: Esteira transportadora de Carvão Mineral

- As linhas são denominadas de TC01, TC02 e TC03 e cada uma tem a capacidade de 2200 t/h.
Com isso, atualmente, é usado como referência uma taxa de eficiência, que é uma média da taxa real que está sendo praticada dividido pelo número de esteiras (taxa de eficiência máxima = 2066t/h)
- Equipamento que permite a modificação no roteamento da carga de carvão dos DNs para os pátios: cabeça móvel. Em princípio, a cabeça móvel não é empregada com frequência tal que uma linha que começa no DN do porto fica automaticamente associada a um ou

mais pátios. Para o caso de quebras ou manutenção de equipamento é possível alterar essa configuração desde que haja um funcionário disponível. Portanto, pode-se concluir que a configuração pode ser modificada se houver uma justificativa para tal. O segundo conjunto de esteiras possui um alcance dos pátios dds por: TC14 {D}; TC12 {B,C}; TC14 {A,G}

- O destino final destas linhas podem ser as Empilhadeiras (EP05, EP06 e EP07), alocadas no pátio de carvão ou direcionar o produto diretamente para o cliente (ArcelorMittal Tubarão), sendo que o cliente recebe apenas por duas linhas.
- A demanda da ArcelorMittal Tubarão gira em torno de 45% da carga movimentada.
- é possível alocar 3 DNs para descarregar um navio de grande porte (CAPESIZE, p.ex.) e 1 DN para um navio de pequeno porte (Handysize).
- Prancha (ritmo) de embarque e descarga se relaciona com o pagamento de multa ou recebimento de prêmio: 50% prêmio, 100% multa. Atualmente o regimento empregado é FIFO. Dessa forma, o problema se torna quais DNs atribuir para cada navio (para a nova configuração do porto não haverá diferença entre os berços em termos de limitação do calado do navio). A programação do descarregamento do navio (ordem de quais porões devem ser descarregados) é acordada entre o planejamento e os armadores do navio.
- CAPESIZE e PANAMAX pagam diferentes valores de demurrage. O Demurrage é estabelecido de acordo com os contratos fechados entre os clientes e os navios e tem por base o uso da taxa de eficiência da prancha.
- A carga de carvão é separada em diferentes porões do navio e possui duas informações que permite a sua diferenciação: tipos de carvão e clientes.
- Atribuição de máquinas (tratores) pode ser realizada em 3 frentes: descarregamento do navio (recheio do porão), pátio (recheio da pilha) e área G (carregamento dos vagões).OBS Fluxo do DN varia de acordo com nível de carvão no porão (necessário recheio em baixo nível)

Abaixo um esquemático do Terminal de Praia Mole:

¹Obs.: os 4 DNs podem operar em apenas um berço, apesar de serem apresentados em berços separados. *Obs2.: Fluxo do DN varia de acordo com nível de carvão no porão (necessário recheio em baixo nível)

1*

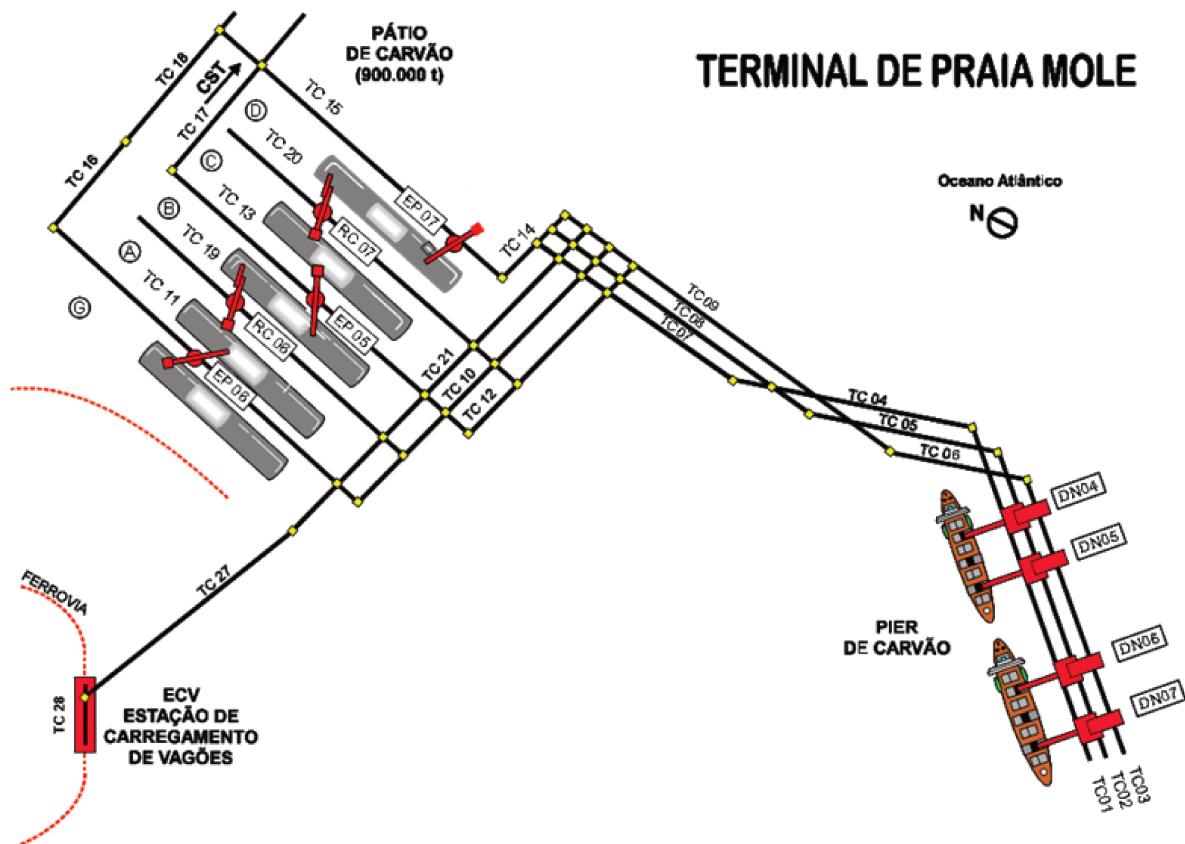


Figura 3.7: Esquemática do Terminal de Praia Mole

3.1.2 Dados do Pátio de Estocagem: quantidade e especificações de recuperadoras, empilhadeiras e pás mecânicas

- Os pátios de estocagem são denominados de área G, A, B, C e D. A área G tem capacidade de estocagem aproximada de 120.000t e as demais possuem uma capacidade de 200.000t. Totalizando 920.000t.
- Para cada cliente e cada tipo de carvão é necessário abrir uma pilha diferente no pátio. Se as pilhas forem de tipos diferentes de carvão, então, é necessário que as pilhas tenham uma distância de 10m para não haver contaminação. Para tanto, é útil observar as marcações no pátio. Cada marcação corresponde a um espaço de 10m.



Figura 3.8: Pátios de Estocagem

- O material estocado na área G é removido apenas por pá carregadeira e expedido por vagões. O material estocado nas demais áreas é removido pelas recuperadoras (RC06 ou RC07) com destino para a ECV (Estação de Carregamento de Vagões) e no momento em que a pilha está sendo finalizada, é utilizada a pá carregadeira para realizar o recheio.



Figura 3.9: Empilhadeira e Recuperadora

- A RC06 tem a capacidade de 2000 t/h, RC07 e as linhas de expedição tem a capacidade de 2200 t/h.
- A ECV tem um silo de armazenamento de $500m^3$ e capacidade de expedição de 2880 t/h.
- 9 pás mecânicas são utilizadas no processo, sendo 3 no Pátio de Carvão, 5 no Píer de Carvão e 1 reserva.

- A política adotada pela empresa para o armazenamento de carvão mineral nos pátios é tal que procura manter o mesmo nível de carvão mineral em todos os pátios.
- As únicas conexões que podem ser alteradas são as dos transportadores (TRTC 07, 08 e 09) para os transportadores (TRTC 10, 12, 21 e 14), sendo que o usual é:
 - TRTC07 x TRTC12
 - TRTC08 x TRTC10
 - TRTC09 x TRTC14
 - A TRTC21 é utilizada apenas no embarque e não no processo de descarga.
- No pátio, os DNs e as EPs podem funcionar simultaneamente no mesmo pátio desde que exista uma distância segura.
- O pedido de um cliente pode ser uma composição de vários tipos de carvão distribuídos em várias pilhas nos pátios. Neste caso, pode ser necessário obedecer a uma pausa para enviar o conteúdo de várias pilhas pela mesma linha.
- Existe uma linha direta para a Arcelor Mittal (pode ser qualquer uma que vem do porto) ou pode-se ainda acionar uma recuperadora (RN) para enviar carvão do pátio para a Arcelor Mittal. As duas coisas podem ocorrer ao mesmo tempo. Transporte de carvão diretamente do navio para a Arcelor Mittal faz com que a empilhadeira da linha usada para o transporte, fique inutilizável.

Para a estocagem de carvão, a empilhadeira posiciona a lança no início do balizamento especificado e sobre o eixo das pilhas do pátio. A lança principia toda baixa e vai subindo (sempre sobre o eixo de empilhamento) à medida que a pilha vai sendo formada. Quando a pilha atinge sua altura máxima, a máquina translada de 2 a 6 metros - valor a ser estabelecido pelo operador, dependendo do tipo do material. A lança abaixa até a menor altura possível sobre a pilha e é continuado o processo, sempre mantendo a base da pilha a aproximadamente três metros da talude, afim de evitar queda de material nas canaletas de drenagem. Conforme a figura 3.10

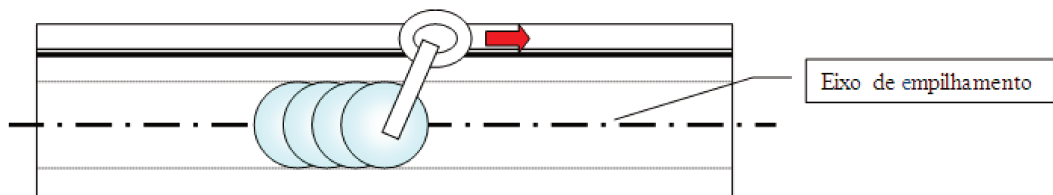


Figura 3.10: Formação da Pilha de Carvão Mineral

Para a estocagem de coque, o empilhamento é realizado em cordões, a máquina translada da baliza inicial à final, gira e forma o segundo cordão paralelamente ao primeiro. Assim ocorre sucessivamente até completar toda a base da pilha, quando então os vãos serão preenchidos na formação dos novos cordões. Conforme a figura 3.11

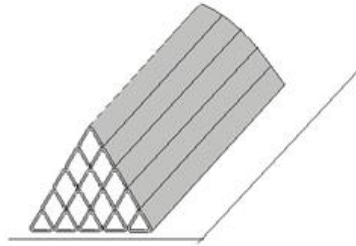


Figura 3.11: Formação da Pilha de Coque

3.1.3 Dados das operações da Estação de Carregamento de Vagões



Figura 3.12: Carregamento de Vagões

- Os vagões posicionados na ECV, são os vagões que chegam no Porto carregados de minério. Os que retornarão carregados com carvão, representam aproximadamente 10% dos que chegam com minério, são limpos e disponibilizados para tal fim.
- Tempos de posicionamento, carregamento e manobras no silo:
 - Na ECV, em média é realizado o carregamento de 8 lotes/dia, sendo que o lote médio é de 84 vagões e eventualmente é carregado um lote com 126 vagões.
 - O silo de carvão opera numa taxa tal que 28 vagões são preenchidos (15 vagões com a capacidade do silo e mais 13 com o envio de carvão da recuperadora) em 10 minutos. Depois disso, é necessário parar para abastecer o silo por 20 minutos. Assim, é necessário realizar duas paradas para completar um lote de 84 vagões.
 - O tempo de carregamento médio é de 1:15h;
 - O tempo de carregamento com zeramento de pilha médio é de 1:50h;

- Após a retirada de um lote da ECV, o tempo acordado com a ferrovia é de 45 min para o posicionamento de novo lote.

Já foi realizado um estudo para a construção de um novo silo, porém o silo existente atenderia a demanda projetada, não sendo viabilizado a construção de um novo silo.

O carvão pode ir direto para a ECV, porém esta alternativa não é usada devido vários fatores, tais como:

- A disponibilidade de carga no Pátio atende a programação desembarque;
- Dificuldade de controlar a quantidade de carga que será enviada para o silo através dos DN's

3.2 Terminal de Produtos Diversos

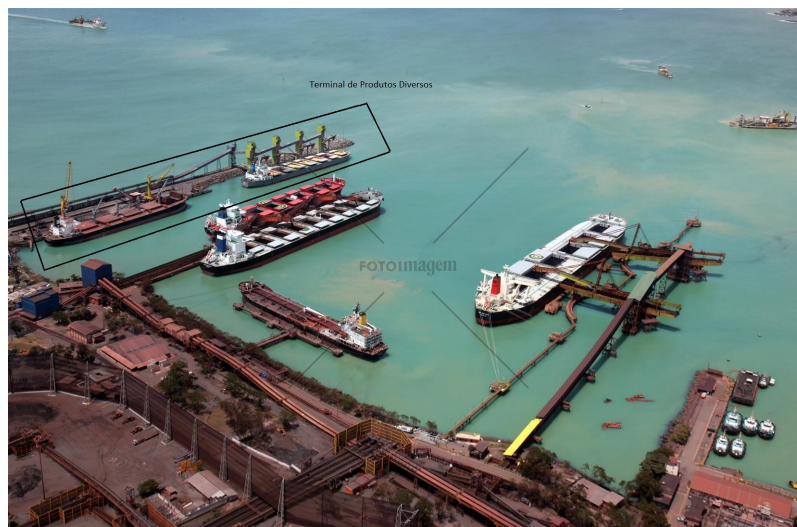


Figura 3.13: Terminal de Produtos Diversos

No terminal de produtos diversos, temos uma restrição de alocação devido ao espaço limitado, quando um navio está atracado no primeiro berço de um dos píeres, não pode haver atracação nem desatracação no segundo berço desse píer. Como ilustra a figura 3.14.

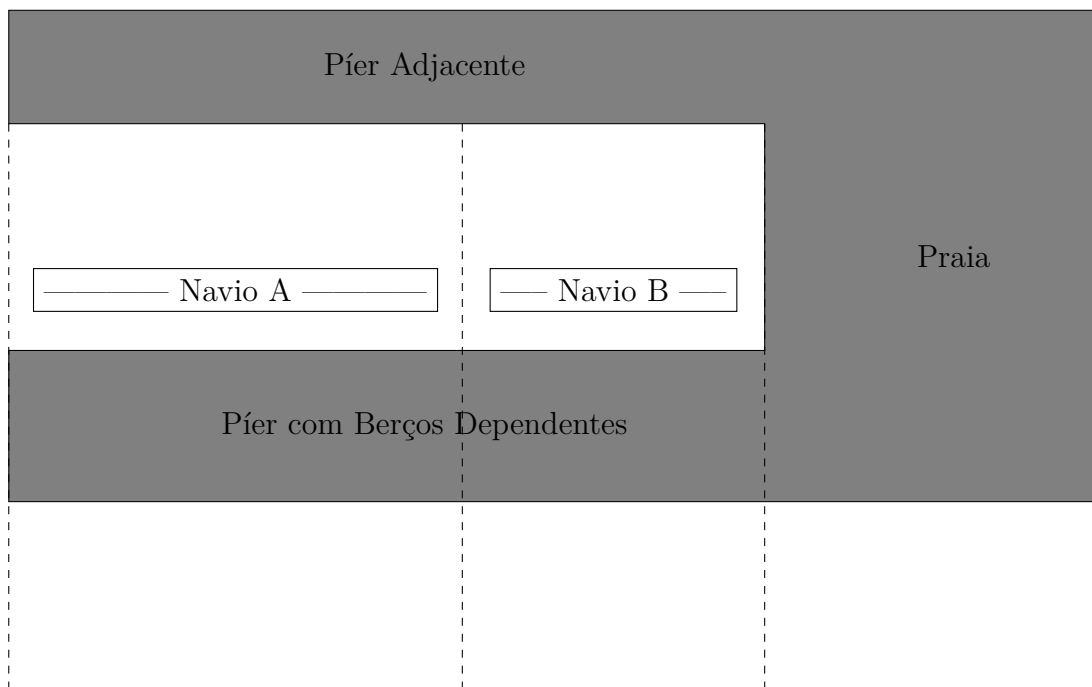


Figura 3.14: Exemplo de berços dependentes

Na figura ?? temos dois píeres nomeados de "Pier com Berços Dependentes" e "Pier Adjacente" entre eles há um corredor com água para a permanência de navios (atracados). Ainda na figura ?? o Berço 2 é dependente do Berço 1, no caso de termos um navio no Berço 1, não pode haver atracação e/ou desatracação no Berço 2, caso não tenha navio no Berço 1, pode haver atracação nem desatracação no Berço 2

Esses e outros processos do porto de Tubarão podem ser encontrados com mais detalhes em Morais [14], da Costa Cruz et al. [3] e Fioroni et al. [5].

Tendo como principal base os dados e processos aqui descritos, vamos trabalhar desde a atracação dos navios nos berços do terminal de praia mole, até a chegada desse carvão mineral nos pátios de armazenamento. Para isso faremos uso das seguintes informações:

- Características dos navios graneleiros de Praia Mole, tais como carga, chegada, número de porões;
- Características dos berços de Praia Mole tais como profundidade disponibilidade de máquinas;
- Características dos descarregadores de navios, tais como berço que pode operar, capacidade, vazão;
- Características das linhas de carvão, tais como capacidade, berço que pode operar;

E para o terminal de Produtos diversos, vamos apenas estudar e criar restrições para representar o problema de berços dependentes.

Capítulo 4

Modelos Matemáticos

Nesse capítulo se encontram os modelos desenvolvidos e usados no projeto.

4.1 Modelagem Matemática para o Terminal de Produtos Diversos

O foco desta pesquisa é o terminal de Praia Mole, contudo alguns resultados foram desenvolvidos para o terminal de Produtos Diversos. Como informado na primeira visita, um dos píeres do terminal de Produtos Diversos possui berços dependentes, devido a esse problema uma nova formulação foi elaborada, com base na formulação (2.1.1).

4.1.1 Atracação em Berços Dependentes

O terminal de Produtos Diversos é tal que a atracação no berço 1 bloqueia a atracação ou a saída no berço 2. Portanto um navio que deseje atracar no berço 2 deve vir antes ou após o(s) navio(s) que atraca(m) no berço 1. Seja um navio i que será atendido pelo berço 2, então um outro navio j da fila de navios pode ter as seguintes opções:

1. O navio j não é atendido pelo berço 1, ou o navio i não é atendido pelo berço 2.
2. O navio j é atendido pelo berço 1 antes do início do atendimento do navio i no berço 2.
3. O navio j é atendido pelo berço 1 após o término do atendimento do navio i no berço 2

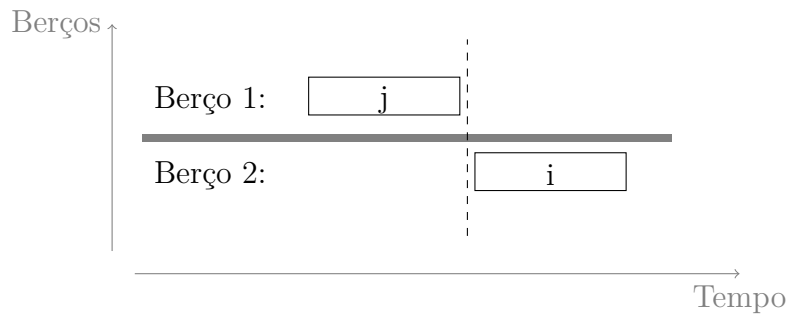


Figura 4.1: Situação 2

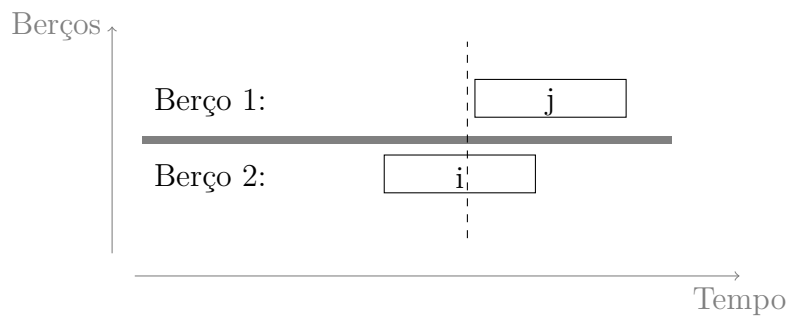


Figura 4.2: Situação 3

Sejam:

Conjuntos

- B - Conjunto dos Berços
- N - Conjunto dos Navios

Variáveis

- $x_{i,j,k} = \begin{cases} 1, & \text{se o navio } i \text{ é atendido após o navio } j \text{ no berço } k \\ 0, & \text{caso contrário.} \end{cases}$
- $T_{i,k}$ - Hora de atracação do navio i no berço k .

Parâmetros

- $t_{i,k}$ - Duração do serviço do navio i no berço k .
- a_i - Chegada do navio i .
- b_i - Término da janela de tempo do navio i .
- M - Número muito grande (relativo aos dados do problema)

- $o(k)$ e $u(k)$ são o primeiro e último navio do berço k , respectivamente.

A restrição de atracação nos berços pode ser expressa em termos matemáticos:

$$T_{i,2} \leq T_{j,1} \quad (\text{primeiro a atracação no berço 2}) \quad (4.1.1a)$$

OU

$$T_{i,2} \geq T_{j,1} + t_{j,1} \quad (\text{só pode atracar no berço 2 depois de liberar o berço 1}) \quad (4.1.1b)$$

Notemos que, as restrições apenas são válidas quando j é atendido no berço 2 e i é atendido no berço 1. Usando esse fato e fazendo uso de técnicas PL para eliminação da conjunção **OU** obtemos o seguinte conjunto de restrições:

$$tda_{i,j} \leq M * y_{i,j} \quad (4.1.2a)$$

$$tda_{i,j} \geq 0 \quad (4.1.2b)$$

$$tdp_{i,j} \geq t_{i,2} * (1 - y_{i,j}) \quad (4.1.2c)$$

$$tdp_{i,j} \leq M * (1 - y_{i,j}) \quad (4.1.2d)$$

$$T_{i,2} + tda_{i,j} = T_{j,1} + tdp_{i,j} \quad (4.1.2e)$$

$$\forall i \in N, \forall j \in N \quad (4.1.2f)$$

Onde

- $y_{i,j}$ - $\begin{cases} 1, & \text{se a restrição (4.1.2a) está ativa} \\ 0, & \text{se a restrição (4.1.2c) e (4.1.2d) estão ativas.} \end{cases}$
- $tda_{i,j}$ - Indica quanto tempo **antes** o navio i foi atendido no berço 2 comparado com o navio j atendido no berço 1.

- $tdp_{i,j}$ - Indica quanto tempo **depois** o navio i foi atendido no berço 2 comparado com o navio j atendido no berço 1.

Caso a restrição ativa seja a (4.1.2a) ($y_{i,k} = 1$), o navio k será atendido pelo berço 2 e o navio i seja atendido pelo berço 1, a restrição se resume a $T_{k,2} + tda_{k,2} \leq T_{i,1}$ indicando que o navio k , atendido no berço 2, terá seu atendimento iniciado antes da atracação do navio i no berço 1, e a restrição (4.1.4p) acompanhado da restrição (4.1.2d) se tornam redundantes quando $y_{i,k} = 0$. Para o caso em que a restrição ativa seja a (4.1.2c) e (4.1.2d) ($y_{i,k} = 0$), o navio k seja atendido pelo berço 2 e o navio i seja atendido pelo berço 1, a restrição se resume a $T_{k,2} \geq T_{i,1} + tdp_{i,1}$ indicando que o navio k , atendido no berço 2, terá seu atendimento iniciado após o termino do atendimento do navio j no berço 1, e a restrição (4.1.2c) se torna redundante quando $y_{i,k} = 0$.

4.1.2 Função Objetivo

A função objetivo dada por $\sum_{i \in N, k \in B} (T_{i,k} - a_i + t_{i,k}) * \sum_{j \in N \cup \{u(k)\}} x_{i,j,k}$ que nada mais é que a somatória dos tempos de espera e dos tempos de atendimento dos navios, é uma equação não linear. O tempo $t_{i,k}$ é variável, pois depende da quantidade de máquinas que atendem o navio i , no berço k , então temos um multiobjetivo agora, que é minimizar o tempo de espera ($T_{i,k} - a_i$) de cada navio no porto, bem como minimizar o seu tempo de atendimento ($t_{i,k}$), para melhor representação do nosso objetivo, vamos multiplicar o tempo de atendimento por um fator $\mu_{i,k} \in [0, 1]$ indicando o peso que cada duração de atendimento tem na função objetivo. Fazendo uso de técnicas de modelagens matemáticas podemos definir uma nova variável $z_{i,k}$ tal que:

$$z_{i,k} \leq M * \sum_{j \in N \cup \{u(k)\}} x_{i,j,k} \quad (4.1.3a)$$

$$z_{i,k} \leq T_{i,k} + \mu_{i,k} t_{i,k} - a_i \quad (4.1.3b)$$

$$z_{i,k} \geq T_{i,k} + \mu_{i,k} t_{i,k} - a_i - M * (1 - \sum_{j \in N \cup \{u(k)\}} x_{i,j,k}) \quad (4.1.3c)$$

$$z_{i,k} \geq 0 \quad (4.1.3d)$$

$$\forall i \in N, \forall k \in B \quad (4.1.3e)$$

Com isso $z_{i,k}$ assume o valor de $(T_{i,k} + \mu_{i,k} t_{i,k}) * \sum_{j \in N \cup \{u(k)\}} x_{i,j,k}$ o problema pode ser formado apenas com restrições lineares.

4.1.3 Formulação Matemática

Nessa subseção se encontra o modelo completo com a adição das restrições (4.1.2) e (4.1.3) construídas acima.

Função Objetivo

$$\min \sum_{i \in N, k \in B} z_{i,k} \quad (4.1.4a)$$

Linearização da Função Objetivo

$$z_{i,k} \leq M * \sum_{j \in N \cup \{u(k)\}} x_{i,j,k} \quad (4.1.4b)$$

$$z_{i,k} \leq T_{i,k} + t_{i,k} - a_i \quad (4.1.4c)$$

$$z_{i,k} \geq T_{i,k} + t_{i,k} - a_i - M * \left(1 - \sum_{j \in N \cup \{u(k)\}} x_{i,j,k}\right) \quad (4.1.4d)$$

O navio deve ser atendido em apenas um berço e apenas uma vez

$$\sum_{j \in N \cup \{o(k), u(k)\}, k \in M} x_{i,j,k} = 1 \quad \forall i \in N \quad (4.1.4e)$$

$$\sum_{i \in N \cup \{o(k), u(k)\}, k \in M} x_{i,j,k} = 1 \quad \forall j \in N \quad (4.1.4f)$$

Garante um primeiro navio em cada berço

$$\begin{aligned} \sum_{j \in N \cup \{o(k), u(k)\}} x_{o(k),j,k} &= 1 \quad \forall k \in B \\ x_{j,o(k),k} &= 0 \quad \forall k \in M, \forall j \in N \cup \{o(k), u(k)\} \end{aligned} \quad (4.1.4g)$$

Garante um último navio em cada berço

$$\begin{aligned} x_{u(k),j,k} &= 0 \quad \forall k \in M, \forall j \in N \cup \{o(k), u(k)\} \\ \sum_{j \in N \cup \{o(k), u(k)\}} x_{j,u(k),k} &= 1 \quad \forall k \in B \end{aligned} \quad (4.1.4h)$$

Continuidade do fluxo de navios

$$\sum_{j \in N \cup \{u(k)\}} x_{i,j,k} - \sum_{j \in N \cup \{o(k)\}} x_{j,i,k} = 0 \quad \forall i \in N, \forall k \in B \quad (4.1.4i)$$

Tempo de atracação maior que de chegada

$$T_{i,k} \geq a_i - M * \left(1 - \sum_{j \in N} x_{i,j,k}\right) \quad \forall i \in N, \forall k \in B \quad (4.1.4j)$$

Término do atendimento antes do limite de tempo

$$T_{i,k} + t_{i,k} \leq b_i + M * \left(1 - \sum_{j \in N} x_{i,j,k} \right) \quad \forall i \in N, \forall k \in B \quad (4.1.4k)$$

Calcula o tempo de atracação

$$T_{i,k} + t_{i,k} - T_{j,k} \leq (1 - x_{i,j,k}) * M \quad \forall i \in N, \forall j \in N, \forall k \in B \quad (4.1.4l)$$

Término do serviço antes do prazo máximo do navio

$$T_{i,k} + t_{i,k} \leq b_i \quad \forall i \in N, \forall k \in B \quad (4.1.4m)$$

Atracação no Berço 2

$$tda_{i,j} \leq M * y_{i,j} \quad (4.1.4n)$$

$$tda_{i,j} \geq 0 \quad (4.1.4o)$$

$$tdp_{i,j} \geq t_{i,1} * (1 - y_{i,j}) \quad (4.1.4p)$$

$$tdp_{i,j} \leq M * (1 - y_{i,j}) \quad (4.1.4q)$$

$$T_{j,2} + tda_{i,j} = T_{i,1} + tdp_{i,j} \quad (4.1.4r)$$

$$\forall i \in N, \forall j \in N \quad (4.1.4s)$$

$$\begin{aligned} y_{i,j} \text{ binário} & \quad \forall i \in N, \forall j \in N \\ x_{i,j,k} \text{ binário} & \quad \forall i \in N, \forall j \in N, \forall k \in B \\ z_{i,k} \geq 0 & \quad \forall i \in N, \forall k \in B \\ T_{i,k} \geq 0 & \quad \forall i \in N, \forall k \in B \end{aligned} \quad (4.1.4t)$$

4.2 Modelagem Matemática para o Terminal de Praia Mole

Vários modelos e casos podem ser avaliados para o terminal de praia mole, que são eles:

1. Alocação das máquinas (descarregadores de navios, esteiras, empilhadeiras e pátios) juntamente com a alocação de navios nos berços.
2. Fixação de descarregadores de navios a determinados berços afim de reduzir a complexidade computacional do modelo.
3. Alocação das máquinas (descarregadores de navios, esteiras, empilhadeiras e pátios), priorizando o atendimento dos navios por FIFO.
4. Fila do tipo FIFO e alocação de máquinas segundo porções.

Trataremos aqui o caso em que a fila é livre com alocação de guindastes e esteiras.

4.2.1 Formulação Considerando Navios e Máquinas

Nessa formulação, consideramos apenas a eficiência do descarregamento (eficiência de prancha), sem considerar o armazenamento dessa carga. Nessa formulação a ordem de atendimento é uma variável de decisão, para avaliar a melhora dos resultados comparado ao atual regime FIFO usado no terminal de Praia Mole.

Nesse caso, para cada navio, não importando em que berço ele estará alocado, devemos disponibilizar ao menos um DN, uma esteira e uma empilhadeira para seu descarregamento. Fixamos as linhas TRTC07 x TRTC12, TRTC08 x TRTC10 e TRTC09 x TRTC14, para reduzir a complexidade do modelo, e para tornar mais praticável a solução do problema.

A alocação das esteiras e guindastes nessa formulação, dá-se de forma semelhante à dos navios. Primeiramente decide-se qual equipamento vai atender determinado navio, depois se calculam os tempos e durações de uso dos equipamentos partindo da sequência de atendimento dos navios pelos equipamentos.

Formulação Matemática

Formulação Matemática para o terminal de Praia Mole considerando a alocação dos navios em berços e das máquinas em navios.

Conjuntos

- B - Conjunto dos Berços.
- N - Conjunto dos Navios.
- D - Conjunto dos DNs.
- E - Conjunto das Esteiras.

Variáveis

- $x_{i,j,k}$ - $\begin{cases} 1, & \text{se o navio } i \text{ é atendido antes do navio } j \text{ no berço } k \\ 0, & \text{caso contrário.} \end{cases}$
- $T_{i,k}$ - Hora de atracação do navio i no berço k .
- $t_{i,k}$ - Duração do serviço do navio i no berço k .
- $y_{i,j,d,k}$ - $\begin{cases} 1, & \text{se o navio } i \text{ é atendido antes do navio } j \text{ pelo descarregador } d \text{ no berço } k \\ 0, & \text{caso contrário.} \end{cases}$
- $Y_{i,d}$ - Hora de início do atendimento do navio i pelo descarregador d .
- $s_{i,j,e,k}$ - $\begin{cases} 1, & \text{se o navio } i \text{ é atendido antes do navio } j \text{ na esteira } e \text{ no berço } k \\ 0, & \text{caso contrário.} \end{cases}$
- $S_{i,e}$ - Hora de início do atendimento do navio i pela esteira e .

- $o(k)$ e $u(k)$ representam o primeiro e último navio do berço k , respectivamente.

Parâmetros

- a_i - Chegada do navio i ao porto.
- Q_i - Carga do navio i .
- tax_d - Taxa de descarga do DN d .
- tax_e - Taxa de descarga da Esteira e .
- M - Número muito grande.

Função Objetivo

$$\min \sum_{i \in N, k \in B} z_{i,k} \quad (4.2.1a)$$

Linearização da Função Objetivo

$$\begin{aligned} z_{i,k} &\leq M * \sum_{j \in NU\{u(k)\}} x_{i,j,k} \\ z_{i,k} &\leq T_{i,k} + t_{i,k} - a_i \\ z_{i,k} &\geq T_{i,k} + t_{i,k} - a_i - M * (1 - \sum_{j \in NU\{u(k)\}} x_{i,j,k}) \end{aligned} \quad (4.2.1b)$$

Duração do Atendimento

$$t_{i,k} \geq \frac{Q_i}{\sum_{j \in NU\{u(k)\}, d \in D} y_{i,j,d,k} * tax_d} \quad \forall i \in N, \forall k \in B \quad (4.2.1c)$$

$$t_{i,k} \geq \frac{Q_i}{\sum_{j \in NU\{u(k)\}, e \in E} s_{i,j,e,k} * tax_e} \quad \forall i \in N, \forall k \in B \quad (4.2.1d)$$

O navio deve ser atendido em apenas um berço e apenas uma vez

$$\sum_{j \in NU\{o(k), u(k)\}, k \in B} x_{i,j,k} = 1 \quad \forall i \in N \quad (4.2.1e)$$

$$\sum_{i \in NU\{o(k), u(k)\}, k \in B} x_{i,j,k} = 1 \quad \forall j \in N \quad (4.2.1f)$$

Garante um primeiro navio em cada berço

$$\begin{aligned} \sum_{j \in NU\{o(k), u(k)\}} x_{o(k),j,k} &= 1 \quad \forall k \in B \\ x_{j,o(k),k} &= 0, \forall j \in N \cup \{o(k), u(k)\}, \forall k \in B \end{aligned} \quad (4.2.1g)$$

Garante um último navio em cada berço

$$\begin{aligned} x_{u(k),j,k} &= 0 & \forall j \in N \cup \{o(k), u(k)\}, \forall k \in B \\ \sum_{j \in N \cup \{o(k), u(k)\}} x_{j,n+1,k} &= 1 & \forall k \in B \end{aligned} \quad (4.2.1h)$$

Continuidade do fluxo de navios

$$\sum_{j \in N \cup \{u(k)\}} x_{i,j,k} - \sum_{j \in N \cup \{o(k)\}} x_{j,i,k} = 0 \quad \forall i \in N, \forall k \in B \quad (4.2.1i)$$

Tempo de atracação maior que de chegada

$$T_{i,k} \geq a_i - M * \left(1 - \sum_{j \in N} x_{i,j,k} \right) \quad \forall i \in N, \forall k \in B \quad (4.2.1j)$$

Cálculo do tempo de atracação

$$T_{i,k} + t_{i,k} - T_{j,k} \leq (1 - x_{i,j,k}) * M \quad \forall i \in N, \forall j \in N, \forall k \in B \quad (4.2.1k)$$

Término do serviço antes do prazo máximo do navio

$$T_{i,k} + t_{i,k} \leq b_i \quad \forall i \in N, \forall k \in B \quad (4.2.1l)$$

Descarregador inicial e final

$$\begin{aligned} \sum_{i \in N \cup \{o(k), u(k)\}} y_{o(k),i,d} &= 1 & \forall d \in Des, \forall k \in B \\ \sum_{i \in N \cup \{o(k), u(k)\}} y_{i,u(k),d} &= 1 & \forall d \in Des, \forall k \in B \\ y_{i,o(k),d} &= 0 & \forall d \in Des, i \in N \cup \{o(k), u(k)\}, \forall k \in B \\ y_{u(k),i,d} &= 0 & \forall d \in Des, i \in N \cup \{o(k), u(k)\}, \forall k \in B \\ y_{i,j,d} &= 0 & \forall d \in Des, i \in N \cup \{o(k), u(k)\}, \forall k \in B \end{aligned} \quad (4.2.1m)$$

Descarregadores por navio

$$\begin{aligned} \sum_{j \in N \cup \{u(k)\}, d \in D} y_{i,j,d,k} &\geq \sum_{j \in N \cup \{u(k)\}} x_{i,j,k} & \forall i \in N \cup \{o(k)\}, \forall k \in B \\ \sum_{j \in N \cup \{u(k)\}, d \in D} y_{i,j,d,k} &\leq 3 * \left(\sum_{j \in N \cup \{u(k)\}} x_{i,j,k} \right) & \forall i \in N \cup \{o(k)\}, \forall k \in B \\ \sum_{i \in N \cup \{o(k)\}, d \in D} y_{i,j,d,k} &\geq \sum_{j \in N \cup \{o(k)\}} x_{i,j,k} & \forall j \in N \cup \{u(k)\}, \forall k \in B \\ \sum_{i \in N \cup \{o(k)\}, d \in D} y_{i,j,d,k} &\leq 3 * \left(\sum_{j \in N \cup \{o(k)\}} x_{i,j,k} \right) & \forall j \in N \cup \{u(k)\}, \forall k \in B \end{aligned} \quad (4.2.1n)$$

Descarregadores presos ao trilho (Limitação em seus movimentos)

$$\begin{aligned} \sum_{j \in NU\{u(k)\}} y_{i,j,05,1} &\leq \sum_{j \in NU\{u(k)\}} y_{i,j,06,1} \leq \sum_{j \in NU\{u(k)\}} y_{i,j,07,1} & \forall i \in N \\ \sum_{j \in NU\{u(k)\}} y_{i,j,06,2} &\leq \sum_{j \in NU\{u(k)\}} y_{i,j,05,2} \leq \sum_{j \in NU\{u(k)\}} y_{i,j,04,2} & \forall i \in N \end{aligned} \quad (4.2.1o)$$

Continuidade do fluxo de descarregadores

$$\sum_{j \in NU\{u(k)\}, k \in B} y_{i,j,d,k} - \sum_{j \in NU\{o(k)\}, k \in B} y_{j,i,d,k} = 0 \quad \forall i \in N, \forall d \in D \quad (4.2.1p)$$

Cálculo do tempo dos descarregadores

$$\begin{aligned} Y_{i,d} + t_{i,k} &\leq Y_{j,d} + M * \left(1 - \sum_{m \in NU\{u(k)\}} y_{i,m,d,k} \right) & \forall i \in N, \forall j \in N \cup \{u(k)\}, \forall d \in D, \forall k \in B \\ Y_{i,d} &\leq T_{i,k} + M * \left(1 - \sum_{j \in NU\{u(k)\}} y_{i,j,d,k} \right) & \forall i \in N, \forall d \in D, \forall k \in B \\ Y_{i,d} &\geq T_{i,k} - M * \left(1 - \sum_{j \in NU\{u(k)\}} y_{i,j,d,k} \right) & \forall i \in N, \forall d \in D, \forall k \in B \end{aligned} \quad (4.2.1q)$$

Esteira inicial e final

$$\begin{aligned} \sum_{i \in NU\{o(k), u(k)\}} s_{o(k),i,e} &= 1 & \forall e \in E, \forall k \in B \\ \sum_{i \in NU\{o(k), u(k)\}} s_{i,u(k),e} &= 1 & \forall e \in E, \forall k \in B \\ s_{i,o(k),e,k} &= 1 & \forall e \in EstA, i \in N \cup \{o(k), u(k)\}, \forall k \in B \\ s_{u(k),i,e,k} &= 1 & \forall e \in EstA, i \in N \cup \{o(k), u(k)\}, \forall k \in B \\ s_{i,i,e,k} &= 1 & \forall e \in EstA, i \in N \cup \{o(k), u(k)\}, \forall k \in B \end{aligned} \quad (4.2.1r)$$

Esteiras por navio

$$\begin{aligned} \sum_{j \in NU\{u(k)\}, e \in E} s_{i,j,e} &\geq \sum_{j \in NU\{u(k)\}} x_{i,j,k} & \forall i \in N \cup \{o(k)\} \\ \sum_{j \in NU\{u(k)\}, e \in E} s_{i,j,e} &\leq 2 * \left(\sum_{j \in NU\{u(k)\}} x_{i,j,k} \right) & \forall i \in N \cup \{o(k)\} \\ \sum_{i \in NU\{o(k)\}, e \in E} s_{i,j,e} &\geq \sum_{i \in NU\{o(k)\}} x_{i,j,k} & \forall j \in N \cup \{u(k)\} \\ \sum_{i \in NU\{o(k)\}, e \in E} s_{i,j,e} &\leq 2 * \left(\sum_{i \in NU\{o(k)\}} x_{i,j,k} \right) & \forall j \in N \cup \{u(k)\} \end{aligned} \quad (4.2.1s)$$

Continuidade do fluxo esteiras

$$\sum_{j \in NU\{u(k)\}} s_{i,j,e} - \sum_{j \in NU\{o(k)\}} s_{j,i,e} = 0 \quad \forall i \in N, \forall k \in B \quad (4.2.1t)$$

Cálculo do tempo para esteira

$$\begin{aligned}
S_{i,d} + t_{i,k} &\leq S_{j,d} + M * \left(1 - \sum_{m \in N \cup \{u(k)\}} s_{i,m,e,k} \right) && \forall i \in N, \forall j \in N \cup \{u(k)\}, \forall e \in E, \forall k \in B \\
S_{i,d} &\leq T_{i,k} + M * \left(1 - \sum_{j \in N \cup \{u(k)\}} s_{i,j,e,k} \right) && \forall i \in N, \forall e \in E, \forall k \in B \\
S_{i,d} &\geq T_{i,k} - M * \left(1 - \sum_{j \in N \cup \{u(k)\}} s_{i,j,e,k} \right) && \forall i \in N, \forall e \in E, \forall k \in B
\end{aligned} \tag{4.2.1u}$$

$$\begin{aligned}
x_{i,j,k} &\text{ binário} && \forall i \in N \cup \{o(k), u(k)\}, j \in N \cup \{o(k), u(k)\}, \forall k \in B \\
T_{i,k} &\geq 0 && \forall i \in N \cup \{o(k), u(k)\}, \forall k \in B \\
z_{i,k} &\geq 0 && \forall i \in N \cup \{o(k), u(k)\}, \forall k \in B \\
y_{i,j,d,k} &\text{ binário} && \forall d \in D, i \in N \cup \{o(k), u(k)\}, j \in N \cup \{o(k), u(k)\}, \forall k \in B \\
s_{i,j,e,k} &\text{ binário} && \forall i \in N \cup \{o(k), u(k)\}, j \in N \cup \{o(k), u(k)\}, e \in E, \forall k \in B \\
S_{i,e} &\geq 0 && \forall i \in N \cup \{o(k), u(k)\}, e \in E
\end{aligned} \tag{4.2.1v}$$

4.2.2 Tempo de Atendimento

O tempo de atendimento dado por (4.2.1c) e (4.2.1d), não é linear. Para reduzir a complexidade da formulação faremos uso de técnicas de programação para linearizar essas restrições. Tomando:

$$t_{i,k} \geq \frac{Q_i}{\sum_{j \in N \cup \{u(k)\}, d \in D} y_{i,j,d,k} * tax_d} \quad \forall i \in N, \forall k \in B \tag{4.2.2a}$$

$$t_{i,k} \geq \frac{Q_i}{\sum_{j \in N \cup \{u(k)\}, e \in E} s_{i,j,e,k} * tax_e} \quad \forall i \in N, \forall k \in B \tag{4.2.2b}$$

Que é equivalente a:

$$\sum_{d \in D} \left(\sum_{j \in N \cup \{u(k)\}} y_{i,j,d,k} * t_{i,k} \right) * tax_d \geq Q_i \quad \forall i \in N, \forall k \in B \tag{4.2.2c}$$

$$\sum_{e \in E} \left(\sum_{j \in N \cup \{u(k)\}} s_{i,j,e,k} * t_{i,k} \right) * tax_e \geq Q_i \quad \forall i \in N, \forall k \in B \tag{4.2.2d}$$

Sejam $p_{i,d,k}$ e $q_{i,e,k}$ tais que:

$$\sum_{d \in D, k \in B} p_{i,d,k} * tax_d \geq Q_i \tag{4.2.2e}$$

$$p_{i,d,k} \leq M * \sum_{j \in N \cup \{u(k)\}} y_{i,j,d,k} \tag{4.2.2f}$$

$$p_{i,d,k} \leq t_{i,k} \quad (4.2.2g)$$

$$p_{i,d,k} \geq t_{i,k} - M * \left(1 - \sum_{j \in NU\{u(k)\}} y_{i,j,d,k}\right) \quad (4.2.2h)$$

$$p_{i,d,k} \geq 0 \quad (4.2.2i)$$

$$\forall i \in N, d \in D, k \in B \quad (4.2.2j)$$

$$\sum_{e \in E, k \in B} q_{i,e,k} * tax_e \geq Q_i \quad (4.2.2k)$$

$$q_{i,e,k} \leq M * \sum_{j \in NU\{u(k)\}} s_{i,j,e,k} \quad (4.2.2l)$$

$$q_{i,e,k} \leq t_{i,k} \quad (4.2.2m)$$

$$q_{i,e,k} \geq t_{i,k} - M * \left(1 - \sum_{j \in NU\{u(k)\}} s_{i,j,e,k}\right) \quad (4.2.2n)$$

$$q_{i,e,k} \geq 0 \quad (4.2.2o)$$

$$\forall i \in N, e \in E, k \in B \quad (4.2.2p)$$

Com isso

$$p_{i,d,k} = \sum_{j \in NU\{u(k)\}} y_{i,j,d,k} * t_{i,k} \quad (4.2.3a)$$

e

$$q_{i,e,k} = \sum_{j \in NU\{u(k)\}} s_{i,j,e,k} * t_{i,k} \quad (4.2.3b)$$

e

$$t_{i,k} = \max \left\{ \frac{Q_i}{\sum_{j \in NU\{u(k)\}, d \in D} y_{i,j,d,k} * tax_d}, \frac{Q_i}{\sum_{j \in NU\{u(k)\}, e \in E} s_{i,j,e,k} * tax_e} \right\} \quad (4.2.3c)$$

Pois tomando $\sum_{j \in NU\{u(k)\}} y_{i,j,d,k} = 0$ ficamos com:

$$p_{i,d,k} \leq 0 \quad (4.2.4a)$$

$$p_{i,d,k} \leq t_{i,k} \quad (4.2.4b)$$

$$p_{i,d,k} \geq -M \quad (4.2.4c)$$

$$p_{i,d,k} \geq 0 \quad (4.2.4d)$$

$$\forall i \in N, d \in D, k \in B \quad (4.2.4e)$$

Indicando que $p_{i,d,k} = 0$.

Para o caso em que $\sum_{j \in N \cup \{u(k)\}} y_{i,j,d,k} = 1$

$$p_{i,d,k} \leq M \quad (4.2.4f)$$

$$p_{i,d,k} \leq t_{i,k} \quad (4.2.4g)$$

$$p_{i,d,k} \geq t_{i,k} \quad (4.2.4h)$$

$$p_{i,d,k} \geq 0 \quad (4.2.4i)$$

$$\forall i \in N, d \in D, k \in B \quad (4.2.4j)$$

Indicando que $p_{i,d,k} = t_{i,k}$. Agora usando o fato de que $\sum_{d \in D, k \in B} p_{i,d,k} * tax_d \geq Q_i$ temos

$$t_{i,k} \geq \frac{Q_i}{\sum_{j \in N \cup \{u(k)\}, d \in D} y_{i,j,d,k} * tax_d} \quad (4.2.4k)$$

O processo é análogo para o cálculo de $t_{i,k}$ segundo a alocação das esteiras.

Os modelos construídos nesse capítulo tem como base modelos clássicos adaptados para as necessidades portuárias. Com os modelos matemáticos definidos, resta comprovar a validade dos mesmos, para isso, no capítulo seguinte, vamos fazer uma série de testes computacionais e discutir a sua aplicabilidade na prática dos e, quando possível, comprá-los com que foi realizado no porto.

Portanto, no modelo, usaremos as seguintes restrições para a representação do tempo de atendimento de cada navio

$$\sum_{d \in D, k \in B} p_{i,d,k} * tax_d \geq Q_i \quad (4.2.5a)$$

$$p_{i,d,k} \leq M * \sum_{j \in N \cup \{u(k)\}} y_{i,j,d,k} \quad (4.2.5b)$$

$$p_{i,d,k} \leq t_{i,k} \quad (4.2.5c)$$

$$p_{i,d,k} \geq t_{i,k} - M * \left(1 - \sum_{j \in N \cup \{u(k)\}} y_{i,j,d,k}\right) \quad (4.2.5d)$$

$$p_{i,d,k} \geq 0 \quad (4.2.5e)$$

$$\forall i \in N, d \in D, k \in B \quad (4.2.5f)$$

$$\sum_{e \in E, k \in B} q_{i,e,k} * tax_e \geq Q_i \quad (4.2.5g)$$

$$q_{i,e,k} \leq M * \sum_{j \in N \cup \{u(k)\}} s_{i,j,e,k} \quad (4.2.5h)$$

$$q_{i,e,k} \leq t_{i,k} \quad (4.2.5i)$$

$$q_{i,e,k} \geq t_{i,k} - M * \left(1 - \sum_{j \in N \cup \{u(k)\}} s_{i,j,e,k}\right) \quad (4.2.5j)$$

$$q_{i,e,k} \geq 0 \quad (4.2.5k)$$

$$\forall i \in N, e \in E, k \in B \quad (4.2.5l)$$

4.3 Heurística da Divisão

Essa Heurística, que foi usada para resolver alguns casos reais da Vale S.A. se mostrou eficiente, apesar de sua simplicidade.

O princípio dessa heurística é a separação dos navios envolvidos no problema em conjuntos com um número menor de navios, e então montamos um problema com cada conjunto de navios, passando como entrada também o horário de abertura dos berços vindos da resolução do problema anterior.

Para que a solução fique o mais próxima possível da solução ótima, deve-se dividir os grupos de navios de tal forma que as chegadas dos navios estejam em um intervalo de tempo muito próximo, dessa forma existe um conflito na atracação desses navios, onde ocorrem os melhores ganhos com o uso de modelos matemáticos para a resolução do PAB.

Seja A o horário que o berço abre e F o horário que o berço fecha, a heurística tem a forma:

Algoritmo 4.1 Heurística da Divisão

Contruir $S = \{s_0, s_1, s_2, \dots\}$ conjuntos de navios.

Definir $F_{s_0}(k) = F(k)$ e $A_{s_0}(k) = A(k)$

for $i = 1, \dots, |S|$ **do**

 Resolver o problema considerando $N = s_i$, $F(k) = F_{s_i}(k)$ e $A(k) = A_{s_i}(k)$

 Definir $F_{s_{i+1}}(k) = F(k)$ e $A_{s_{i+1}}(k) = \max_j \{T_{j,k} + t_j, k\}$.

end for

Retornar as soluções obtidas.

Capítulo 5

Testes e Resultados

Nesse Capítulo são apresentadas alguns testes e resultados obtidos com a implementação dos modelos do Capítulo 4.

5.1 Testes

O objetivo desta seção é validar os modelos propostos, para isso faremos uso do software FICOTM *Xpress Optimization Suite* que é uma ferramenta com o pacote CPLEX capaz de resolver problemas de grande escala e complexos com milhões de variáveis e restrições. Os casos aqui testados são de pequena escala, pois até mesmo o problema de alocação de navios em berços, que é um problema de complexidade menor que as modelagens propostas, precisa de uma heurística para resolução de casos grandes, devido ao tempo inviável de resolução em softwares de otimização.

5.1.1 Testes para o Terminal de Produtos Diversos

Caso Teste 1: Caso1PD

Este caso chamado de caso Caso1PD, é um caso muito simples, com apenas 4 navios, onde a solução pode ser encontrada analiticamente.

- 4 navios $N : [1\ 2\ 3\ 4]$
- 2 berços $B : [1\ 2]$
- Os 4 navios já estão no porto $CHEGADA : [0\ 0\ 0\ 0]$

- Tempos de atendimento $t:NxB\ t = \begin{bmatrix} 7 & 7 \\ 7 & 7 \\ 7 & 7 \\ 7 & 7 \end{bmatrix}$

- Valor de M (número grande) $M : 9000$

- Primeiro navio de cada berço $o : 0$
- último navio de cada berço $u : 5$.

Navios	N1	N2	N3	N4
Chegada (h)	0	0	0	0
$t_{i,1}$ (hs)	7	7	7	7
$t_{i,2}$ (hs)	7	7	7	7

Tabela 5.1: Dados do Caso1PD

Caso Teste 2: Caso2PD

Este caso chamado de caso Caso2PD, é um caso simples, com 6 navios, onde a solução pode ser encontrada analiticamente.

- 6 navios $N : [1 \ 2 \ 3 \ 4 \ 5 \ 6]$
- 2 berços $B : [1 \ 2]$
- Os 6 navios já estão no porto $CHEGADA : [0 \ 0 \ 0 \ 0 \ 0 \ 0]$

- Tempos de atendimento $t:N \times B \ t = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 2 & 1 \\ 1 & 2 \\ 2 & 1 \\ 1 & 2 \end{bmatrix}$

- Valor de M (número grande) $M : 9000$
- Primeira navio de cada berço $o : 0$
- último navio de cada berço $u : 7$.

Navios	N1	N2	N3	N4	N5	N6
Chegada (h)	0	0	0	0	0	0
$t_{i,1}$ (hs)	2	1	2	1	2	1
$t_{i,2}$ (hs)	1	2	1	2	1	2

Tabela 5.2: Dados do Caso2PD

Caso Teste 3: Caso3PD

Este caso chamado de caso Caso3PD, é um caso fictício, com 5 navios.

- 5 navios $N : [1\ 2\ 3\ 4\ 5]$
- 2 berços $B : [1\ 2]$
- Tempo de chegada de cada um dos navios $CHEGADA : [4\ 5\ 4\ 8\ 7]$

- Tempos de atendimento $t:NxB\ t = \begin{bmatrix} 2 & 2 \\ 5 & 6 \\ 6 & 3 \\ 5 & 4 \\ 6 & 5 \end{bmatrix}$

- Valor de M (número grande) $M : 9000$
- Primeiro navio de cada berço $o : 0$
- último navio de cada berço $u : 6$.

Navios	N1	N2	N3	N4	N5
Chegada (h)	4	5	4	8	7
$t_{i,1}$ (hs)	2	5	6	5	6
$t_{i,2}$ (hs)	2	6	3	4	5

Tabela 5.3: Dados do Caso3PD

Caso Teste 4: ValePD

Este caso chamado de caso ValePD, é um caso baseado nos dados fornecidos pela empresa Vale S.A..

- 12 navios $N : [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12]$
- 2 berços $B : [1\ 2]$
- Tempo de chegada de cada um dos navios
 $CHEGADA : [0,00\ 0,00\ 0,00\ 0,00\ 4,75\ 6,92\ 0,00\ 20,15\ 16,58\ 8,33\ 11,61\ 0,00]$

- Tempos de atendimento $t:NxB$ $t = \begin{bmatrix} 2,73 & 6,99 \\ 0,00 & 0,01 \\ 0,44 & 1,13 \\ 1,81 & 4,64 \\ 0,70 & 1,78 \\ 2,32 & 5,93 \\ 2,09 & 5,33 \\ 0,13 & 0,34 \\ 0,00 & 0,00 \\ 2,37 & 6,05 \\ 2,55 & 6,51 \\ 0,32 & 0,81 \end{bmatrix}$

- Valor de M (número grande) $M : 1000$
- Primeiro navio de cada berço $o : 0$
- último navio de cada berço $u : 13$.

Navios	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12
Chegada (h)	0,00	0,00	0,00	0,00	4,75	6,92	0,00	20,15	16,58	8,33	11,61	0,00
$t_{i,1}$ (hs)	2,37	0,00	0,44	1,81	0,70	2,32	2,09	0,13	0,00	2,37	2,55	0,32
$t_{i,2}$ (hs)	6,99	0,01	1,13	4,64	1,78	5,93	5,33	0,34	0,00	6,05	6,51	0,81

Tabela 5.4: Dados do Caso ValePD

5.1.2 Testes para Praia Mole

Caso Teste 1: Caso1PM

Este caso chamado de caso Caso1PM, é um caso muito simples, com apenas 3 navios, onde a solução pode ser encontrada analiticamente.

- 3 navios $N : [1 \ 2 \ 3]$
- 2 berços $B : [1 \ 2]$
- 4 descarregadores de navios $D : [DN07 \ DN06 \ DN05 \ DN04]$
- 3 esteiras $E : [TC01 \ TC02 \ TC03]$
- Os 3 navios já estão no porto $CHEGADA : [0 \ 0 \ 0]$
- Carga de cada navio $CARGA : [22000 \ 18000 \ 18000]$
- Vazão de cada um dos DN's $VAZAO_{DESCARREGADOR} : [2000 \ 2000 \ 1800 \ 1800]$

- Vazão de cada uma das esteiras $VAZAO_{ESTEIRA} : [2200 \ 2200 \ 2200]$
- Valor de M (número grande) $M : 9000$
- Primeiro navio de cada berço $o : 0$
- último navio de cada berço $u : 4$
- Peso do tempo de atendimento $\mu = 1$.

Navios	N1	N2	N3
Chegada (h)	0	0	0
Carga (tons)	22000	18000	18000

Descarregadores	DN04	DN05	DN06	DN07
Vazão (tons/h)	2000	2000	1800	1800

Esteiras	TC01	TC02	TC03
Vazão(tons/h)	2200	2200	2200

Tabela 5.5: Dados do Caso1PM

Caso Teste 2: Caso2PM

Este caso chamado de caso Caso2PM, é um caso fictício, com 4 navios.

- 4 navios $N : [1 \ 2 \ 3 \ 4]$
- 2 berços $B : [1 \ 2]$
- 4 descarregadores de navios $D : [DN07 \ DN06 \ DN05 \ DN04]$
- 3 esteiras $E : [TC01 \ TC02 \ TC03]$
- Tempo de chegada de cada um dos navios $CHEGADA : [10 \ 5 \ 4 \ 8]$
- Carga de cada navio $CARGA : [22000 \ 18000 \ 18000 \ 22000]$
- Vazão de cada um dos DN's $VAZAO_{DESCARREGADOR} : [2000 \ 2000 \ 1800 \ 1800]$
- Vazão de cada uma das esteiras $VZSAO_{ESTEIRA} : [2200 \ 2200 \ 2200]$
- Valor de "M" (número grande) $M : 9000$
- Primeiro navio de cada berço $o : 0$
- último navio de cada berço $u : 5$
- Peso do tempo de atendimento $\mu = 1$.

Navios	N1	N2	N3	N4
Chegada (h)	10	5	4	8
Carga (tons)	22000	18000	18000	22000

Descarregadores	DN04	DN05	DN06	DN07
Vazão (tons/h)	2000	2000	1800	1800

Esteiras	TC01	TC02	TC03
Vazão(tons/h)	2200	2200	2200

Tabela 5.6: Dados do Caso2PM

Caso Teste 3: ValePM

Este caso chamado de caso ValePM, é um caso baseado nos dados mensais fornecidos pela empresa Vale S.A..

- 12 navios N : [1 2 3 4 5 6 7 8 9 10 11 12]
- 2 berços B : [1 2]
- 4 descarregadores de navios D : [DN07 DN06 DN05 DN04]
- 3 esteiras E : [TC01 TC02 TC03]
- Tempo de chegada de cada um dos navios
 $CHEGADA$: [0 4 5, 5 8, 95 11, 05 12, 30 14, 42 12, 79 16, 29 19, 33 22, 33 24, 75]
- Carga de cada navio
 $CARGA$: [71436, 93 70694, 48 30540 10810 25697 43991 30000 79996 72549 72303 76989 79648]
- Vazão de cada um dos DN's $VAZAO_{DESCARREGADOR}$: [9600 9600 8640 8640]
- Vazão de cada uma das esteiras $VAZAO_{ESTEIRA}$: [10560 10560 10560]
- Valor de M (número grande) M : 20
- Peso do tempo de atendimento $\mu = 1$.

Navios	N1	N2	N3	N4	N5	N6	N7	N8	N9
Chegada (d)	0	4	5,5	8,95	11,05	12,30	14,42	12,79	16,29
Carga (tons)	71436,93	70694,48	30540	10810	25697	43991	30000	79996	72549
Navios	N10	N11	N12						
Chegada (d)	19,33	22,33	24,75						
Carga (tons)	72303	76989	79648						
Descarregadores	DN04	DN05	DN06	DN07					
Vazão (tons/h)	9600	9600	8640	8640					
Esteiras	TC01	TC02	TC03						
Vazão(tons/d)	10560	10560	10560						

Tabela 5.7: Dados do CasoValePM

Caso Teste 4: ValePMn

Este caso chamado de caso ValePMn, é um caso baseado em dados detalhados fornecidos pela empresa Vale S.A..

- 6 navios N : [1 2 3 4 5 6]
- 2 berços B : [1 2]
- 4 descarregadores de navios D : [DN07 DN06 DN05 DN04]
- 3 esteiras E : [TC01 TC02 TC03]
- Tempo de chegada de cada um dos navios $CHEGADA$: [23, 38 26, 85 29, 65 22, 49 25, 38 29, 06]
- Carga de cada navio $CARGA$: [77042 72895 33644 72349 76600 72240]
- Vazão de cada um dos DN's $VAZAO_{DESCARREGADOR}$: [36000 36000 32400 32400]
- Vazão de cada uma das esteiras $VAZAO_{ESTEIRA}$: [39840 39840 39840]
- Valor de "M" (número grande) M : 6
- Peso do tempo de atendimento $\mu = 0,5$.

Navios	N1	N2	N3	N4	N5	N6
Chegada (d)	23,38	26,85	29,65	22,49	25,38	29,06
Carga (tons)	77042	72895	33644	72349	76600	72240

Descarregadores	DN04	DN05	DN06	DN07
Vazão (tons/d)	36000	36000	32400	32400

Esteiras	TC01	TC02	TC03
Vazão(tons/d)	39840	39840	39840

Tabela 5.8: Dados do Caso ValePMn

5.2 Resultados

Nessa seção estão os resultados obtidos com as formulações elaboradas. Os testes foram feitos em um notebook workstation Lenovo ThinkPad série w510, com um processador core i7 1ª geração Q720 1.6GHz, com 8Gb de memória RAM, com o sistema operacional e windows 8.

5.2.1 Resultados do Terminal de Produtos Diversos

Caso1PD

Nesse caso, que é um caso de pequena escala e de fácil resolução, com apenas 116 restrições, 90 variáveis e 484 elementos não nulos, a solução ótima foi encontrada em 0.2 segundos.

Berço 1	N3	N2
Atracação	0,00	7,00
Serviço	7,00	7,00
Berço 2	N4	N1
Atracação	0,00	7,00
Serviço	7,00	7,00

Tabela 5.9: Resultados do Caso1PD

Caso1PD		
Berço 1	N3	N2
Berço 2	N4	N1

Essa solução possui um *gap* relativo de 0%, indicando que é uma solução ótima do problema, facilmente podemos verificar que essa solução é a melhor solução factível para o problema. Qualquer alocação em sequência é uma solução para esse problema. Portanto concluímos que a solução é ótima.

Caso2PD

Nesse caso, que é um caso de pequena escala e de fácil resolução, com apenas 232 restrições, 182 variáveis e 1060 elementos não nulos, a solução ótima foi encontrada em 2 segundos.

Berço 1	N4	N6	N2
Atracação	0,00	1,00	2,00
Serviço	1,00	1,00	1,00
Berço 2	N5	N1	N3
Atracação	0,00	1,00	2,00
Serviço	1,00	1,00	1,00

Tabela 5.10: Resultados do Caso2PD

Caso2PD			
Berço 1	N4	N6	N2
Berço 2	N5	N1	N3

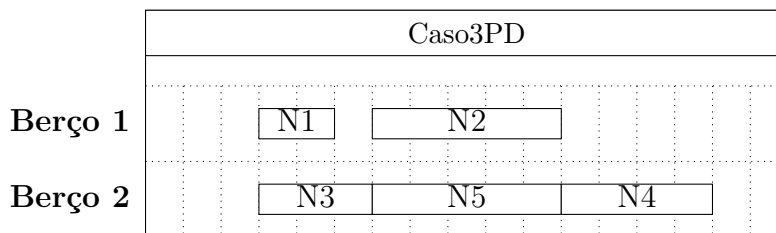
Essa solução possui um *gap* relativo de 0%, indicando que é uma solução ótima do problema, facilmente podemos verificar que essa solução é a melhor solução factível para o problema. Esse caso tem uma solução ótima que pode ser achada facilmente analiticamente, que é alocar os navios pares no berço 1 e os navios ímpares no berço 2, já que os navios pares possuem tempo de atendimento menos no berço 1 e maior no berço 2, e os ímpares tem menor tempo no berço 2 e maior no berço 1, e todos eles estão no porto. Portanto concluímos que a solução obtida é ótima.

Caso3PD

Nesse caso, que é um caso de pequena escala e de fácil resolução, com apenas 134 restrições, 99 variáveis e 964 elementos não nulos, a solução ótima foi encontrada em aproximadamente 1 segundo.

Berço 1	N1	N2	
Atracação	4,00	7,00	
Serviço	2,00	5,00	
Berço 2	N3	N5	N4
Atracação	4,00	7,00	12,00
Serviço	3,00	5,00	4,00

Tabela 5.11: Resultados do Caso3PD

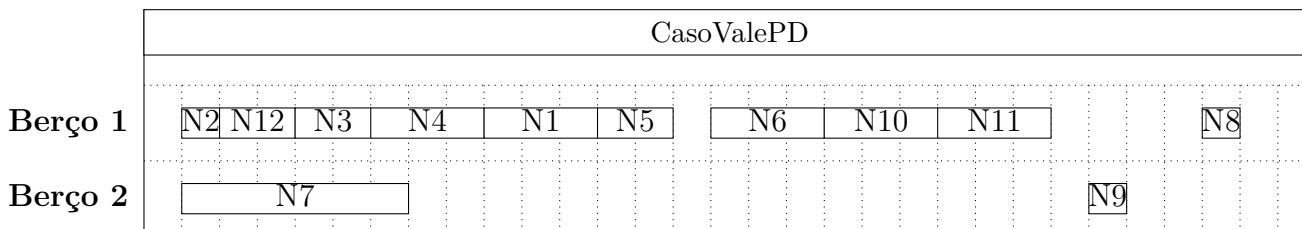


Essa solução possui um *gap* relativo de 0%, indicando que é uma solução ótima do problema.

ValePD

Nesse caso considerado de grande escala, com 592 restrições, 486 variáveis e 2948 elementos não nulos. O software foi processado até que encontrasse uma solução factível ótima, que ocorreu em 436.2 segundos .

Berço 1	N2	N12	N3	N4	N1	N5	N6	N10	N11	N8
Atracação	0,00	0,00	0,32	0,76	2,57	5,3	6,92	9,24	11,61	20,15
Serviço	0,00	0,32	0,44	1,81	2,73	0,70	2,32	2,37	2,55	0,13
Berço 2		N7	N9							
Atracação		0,00	16,58							
Serviço		5,33	0,00							



Essa solução possui um *gap* relativo de 0.00%, indicando que é uma solução ótima do problema, facilmente podemos verificar que essa solução é uma solução factível. OBS: A figura não está em escala, deve ser usada apenas como referência do sequenciamento de navios.

5.2.2 Resultados Praia Mole

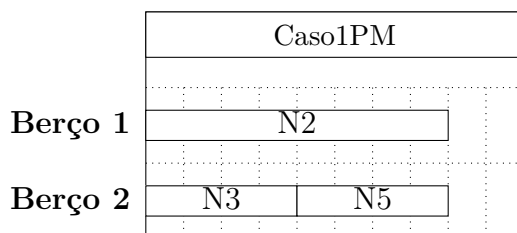
Nesta subseção estão os resultados de cada um dos testes rodados para a formulação de Praia Mole (4.2.1). Vale lembrar que nos casos dessa subseção não temos restrição de atracação nos berços.

Caso Caso1PM

Nesse caso, que é um caso de pequena escala e de fácil resolução, com apenas 335 restrições, 253 variáveis e 2436 elementos não nulos, a solução ótima foi encontrada em 1.2 segundos.

Berço 1	N2	
Atracação	0,00	
Serviço	9,00	
DNs	(07)	
Esteiras	(03)	
Berço 2	N3	N1
Atracação	0,00	4,09
Serviço	4,09	5,00
DNs	(04,05,06)	(04,05,06)
Esteiras	(01,02)	(01,02)

Tabela 5.12: Resultados do Caso Caso1PM



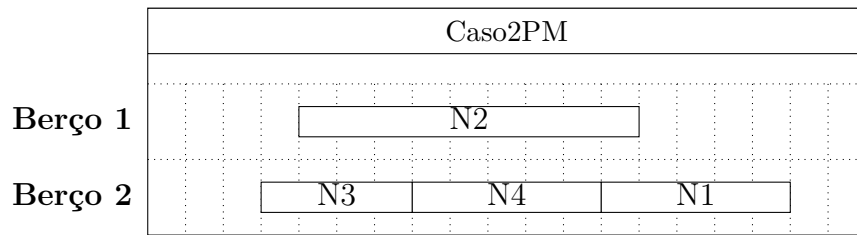
Essa solução possui um *gap* relativo de 0%, indicando que é uma solução ótima do problema, facilmente podemos verificar que essa solução é a melhor solução factível para o problema.

Caso Caso2PM

Nesse caso, que é um caso de pequena escala e de fácil resolução, com apenas 662 restrições, 444 variáveis e 4540 elementos não nulos, a solução ótima foi encontrada em 350.1 segundos.

Berço 1	N2		
Atracação	5,00		
Serviço	9,00		
DNs	(07)		
Esteiras	(02)		
Berço 2	N3	N4	N1
Atracação	4,00	8,09	13,09
Serviço	4,09	5,00	5,00
DNs	(04,05,06)	(04,05,06)	(04,05,06)
Esteiras	(01,03)	(01,03)	(01,03)

Tabela 5.13: Resultados do Caso2PM



Essa solução possui um *gap* relativo de 0%, indicando que é uma solução ótima do problema.

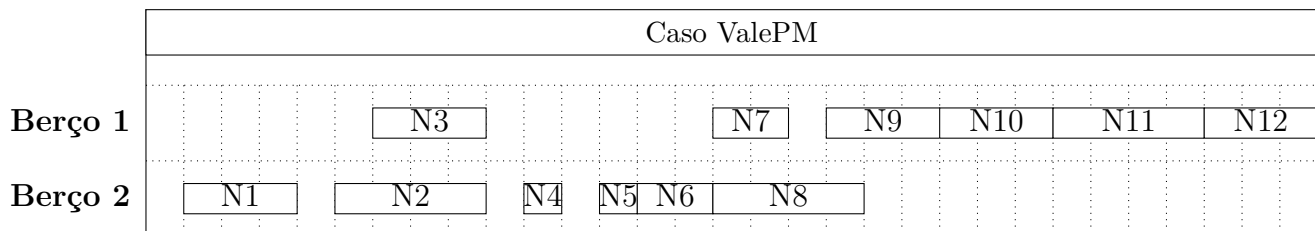
Caso ValePM

Nesse caso, uma heurística bem simples foi usada para a resolução desse caso, a solução foi encontrada em cerca de 130 segundos. A heurística está descrita no apêndice.

Berço 1	N3	N7	N9	N10	N11	N12
Atracação	5,5	14,42	16,99	19,74	22,48	25,39
Serviço	3,18	2,5	2,75	2,74	2,92	3,02
DNs	(07)	(07)	(07,06,05)	(07,06,05)	(07,06,05)	(07,06,05)
Esteiras	(02)	(01)	(01,02)	(01,02)	(01,02)	(01,02)

Berço 2	N1	N2	N4	N5	N6	N8
Atracação	0	4	8,95	11,05	12,3	13,96
Serviço	3,38	3,34	0,51	0,97	1,66	3,03
DNs	(04,05,06)	(04,05,06)	(04,05,06)	(04,05,06)	(04,05,06)	(04,05,06)
Esteiras	(01,03)	(01,03)	(01,03)	(02,03)	(02,03)	(02,03)

Tabela 5.14: Resultados do Caso ValePM



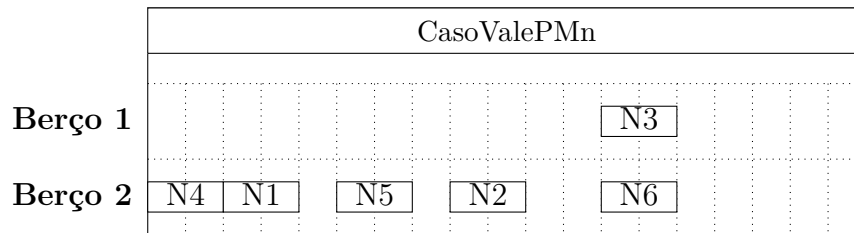
OBS: A figura não está em escala, deve ser usada apenas como referência do sequenciamento de navios.

Caso ValePMn

Nesse caso uma solução ótima foi encontrada em 2952.2 segundos.

		Berço 1		N3	
		Atracação		29,65	
		Serviço		1,45	
		DNs		(07)	
		Esteiras		(01)	
Berço 2	N4	N1	N5	N2	N6
Atracação	22,49	23,86	25,38	26,85	29,06
Serviço	1,37	1,46	1,45	1,38	1,37
DNs	(04,05,06)	(04,05,06)	(04,05,06)	(04,05,06)	(04,05,06)
Esteiras	(02,03)	(02,03)	(02,03)	(02,03)	(02,03)

Tabela 5.15: Resultados do Caso ValePMn



OBS: A figura não está em escala, deve ser usada apenas como referência do sequenciamento de navios.

Tomemos este caso apresentado, dividiremos em 2 grupos: $\{1\ 2\ 4\ 5\}$ e $\{3,6\}$ e usaremos a heurística proposta. A solução foi obtida em 8,6 segundos e é idêntica a solução da otimização sem a divisão de grupos.

		Berço 1		N3	
		Atracação		29,65	
		Serviço		1,45	
		DNs		(07)	
		Esteiras		(01)	
Berço 2	N4	N1	N5	N2	N6
Atracação	23,38	25,38	26,85	29,06	29,06
Serviço	1,37	1,46	1,45	1,38	1,37
DNs	(04,05,06)	(04,05,06)	(04,05,06)	(04,05,06)	(04,05,06)
Esteiras	(02,03)	(02,03)	(02,03)	(02,03)	(02,03)

Aqui apresentamos o que foi realizado pela Vale S.A. para o descarregamento desses navios.

Berço 1	N4	N5	N6
Atracação	22,58	27,52	30,13
Serviço	4,75	2,38	1,78
DNs	(04,05,06*)	(04,05)	(04,05,06*)
Esteiras	(01*,03)	(03)	(01)
Berço 2	N1	N2	N3
Atracação	23,58	28,31	30,95
Serviço	2,14	4,52	2,13
DNs	(06,07)	(06,07)	(04*,05,06,07)
Esteiras	(01*,02)	(01,02)	(01,02,03)

é possível notar que a um revesamento de máquinas nos navios, bem como várias paradas programadas tais como troca de turno, manutenção preventiva, troca de porões, atracação e desatracação de navios, dentre outros, ou não programadas tais como mau tempo, quebra de equipamentos, acidentes manutenções de equipamentos, dentre outros durante o serviço de cada navio.

Capítulo 6

Conclusões e Perspectivas

Os resultados obtidos mostram que as modelagens propostas para os terminais de Produtos Diversos e de Praia Mole representam bem as operações atualmente empregadas nesses terminais e podem constituir ferramentas úteis de otimização. É importante ressaltar que as modelagens desse trabalho não possuem precedentes, e podem vir a ser uma importante ferramenta para o planejamento portuário pois nenhuma ferramenta que faça uso de otimizadores possui um grau de detalhamento similar ao mostrado nesse trabalho. Os modelos matemáticos podem ser usado para criação de *softwares* para planejar melhor o sequenciamento de navios, o funcionamento dos berços, avaliar a possibilidade de construção de novos berços, a aquisição de novas máquinas portuárias, dentre outros. Em particular pode ser usado nos portos do Brasil que possuem investimentos escassos em tecnologia portuária, o que torna imprescindível um planejamento estratégico para aproveitar ao máximo os atuais recursos disponíveis, melhorando assim a competitividade dos portos brasileiros com relação aos portos dos demais países apesar da falta de investimentos.

Essas formulações, assim como as formulações clássicas para o PAB, possuem um custo computacional muito grande, porém sua originalidade e utilidade viabilizam seu uso, e como mostrado aqui, com uma heurística simples adequada a cada porto em questão, é possível obter soluções de cenários reais em tempos computacionais muito baixos e portanto altamente viáveis.

Os modelos podem ser fácil mente adaptados para serem empregados em qualquer porto, pois trabalha com princípio de sequenciamento de máquinas, então basta apenas trocar os nomes e criar novas variáveis conforme o tipo e quantidade de máquinas operando no cais. É possível também realizar algumas simplificações e especializações nos modelos para melhorar sua complexidade computacional sem grandes perdas na qualidade da solução.

Devido a falta de dados precisos como tempo de atracação, tempo de troca de porão, taxa de descarga dos DN's em porões a baixo nível, dentre outros dados, os tempos de preparação (*setups*) foram estimados diretamente na taxa diária de descarga das máquinas no terminal de Praia Mole, onde também foram obtidos bons resultados.

Referências Bibliográficas

- [1] Cordeau, J.-F., Laporte, G., Legato, P. and Moccia, L. [2005], ‘Models and tabu search heuristics for the berth-allocation problem’, *Transportation science* **39**(4), 526–538.
- [2] CPLEX, I. [2007], ‘11.0 user’s manual’, *ILOG SA, Gentilly, France* .
- [3] da Costa Cruz, M. M., Farias, F. N., de Souza, R. G., Vaz, R. G. and da Silva, S. [n.d.], ‘Estudo por simulação da operação de graneis sólidos do porto de tubarão’.
- [4] Feo, T. A. and Resende, M. G. [1989], ‘A probabilistic heuristic for a computationally difficult set covering problem’, *Operations research letters* **8**(2), 67–71.
- [5] Fioroni, M. M., Botter, P. D. R. C., Pompermayer, F. M. and Franzese, L. A. G. [2004], ‘Estudo estratégico de dimensionamento da infra-estrutura do pátio ferroviário de tubarão através de simulação’, *Anais do XVIII ANPET* pp. 8–12.
- [6] Gilmore, P. C. and Gomory, R. E. [1961], ‘A linear programming approach to the cutting-stock problem’, *Operations research* **9**(6), 849–859.
- [7] Glover, F. [1996], ‘Ejection chains, reference structures and alternating path methods for traveling salesman problems’, *Discrete Applied Mathematics* **65**(1), 223–253.
- [8] Imai, A., Nishimura, E. and Papadimitriou, S. [2001], ‘The dynamic berth allocation problem for a container port’, *Transportation Research Part B: Methodological* **35**(4), 401–417.
- [9] Imai, A., Nishimura, E. and Papadimitriou, S. [2003], ‘Berth allocation with service priority’, *Transportation Research Part B: Methodological* **37**(5), 437–457.
- [10] Jarvis, M. S. B. J. J. and Serali, H. D. [2004], *Linear Programming and Network Flows*, Wiley-Interscience.
- [11] Land, A. H. and Doig, A. G. [1960], ‘An automatic method of solving discrete programming problems’, *Econometrica: Journal of the Econometric Society* pp. 497–520.
- [12] Lopes, A. T., Schulz, V. M. L. and Mauri, G. R. [2011], ‘Grasp com path relinking para o problema de alocação de berços’, *Pesquisa Operacional para o Desenvolvimento* **3**(3), 218–229.
- [13] Mitchell, J. E. [2002], ‘Branch-and-cut algorithms for combinatorial optimization problems’, *Handbook of Applied Optimization* pp. 65–77.

- [14] Morais, M. A. C. [2001], ‘Análise da operacionalidade do terminal de produtos siderúrgicos do porto de praia mole: Comparativo de resultados da teoria de filas e simulação computacional’, *Monografia do Curso de Especialização Logística e Transportes do Departamento de Engenharia de Produção da Universidade Federal do Espírito Santo* .
- [15] Nemhauser, G. and Wolsey, L. [1988], *Integer and combinatorial optimization*, Vol. 18, Wiley New York.
- [16] Render, B., Stair, R. M., Hanna, M. E., Wang, M. and Huang, T. [1991], *Quantitative analysis for management*, Allyn and Bacon.
- [17] Xpress, F. [2011], ‘User guide (xpress-ive version 7.2. 1)’, *Mathematical Program Software Manual* .

Apêndice A

Códigos Usados

Nesse apêndice estão os códigos em Xpress usados para a resolução dos casos testes descritos no Capítulo 5.

A.1 Terminal de Produtos Diversos

```
1 model TPD
2 uses "mmxprs"; !gain access to the Xpress-Optimizer solver
3
4 !sample declarations section
5 declarations
6 ! Conjuntos
7     Ber: set of real
8     Nav: set of real
9     Navi: set of real
10    Navf: set of real
11    Navif: set of real
12    Des: set of string
13    EstA: set of string
14    EstB: set of string
15
16
17 ! Variaveis
18     x: array(Navif, Navif, Ber) of mpvar
19     T: array(Navf, Ber) of mpvar
20     z: array(Navf, Ber) of mpvar
21     y: array(Nav, Nav) of mpvar
22     tda: array(Nav, Nav) of mpvar
23     tdp: array(Nav, Nav) of mpvar
24
```

```

25 ! Parametros
26     Cheg: array(Nav) of real
27     Carg: array(Nav) of real
28     M: real
29     t: array(Nav, Ber) of real
30     u: real
31 end-declarations
32
33 initializations from 'ValeCG1.dat'
34     Ber as 'BERCOS'
35     Nav as 'NAVIOS'
36     Navi as 'NAVIOSi'
37     Navf as 'NAVIOSf'
38     Navif as 'NAVIOSif'
39     M as 'M'
40     Cheg as 'CHEGADA'
41     t as 't'
42     u as 'u'
43 end-initializations
44
45 obj:= sum(i in Nav, k in Ber)z(i,k)
46 !obj:= sum(i in Nav, k in Ber)(T(i,k)+t(i,k)) !+ sum(k in Ber)(T(u,
    k)) !+ 0.1*sum(i in Nav,k in Nav)(tda(i,k) + tdp(i,k))
47
48 forall(i in Nav, k in Ber) do
49     z(i,k) <= M*sum(j in Navf)x(i,j,k)
50     z(i,k) <= T(i,k)+t(i,k)-Cheg(i)
51     z(i,k) >= T(i,k)+t(i,k)-Cheg(i)- M*(1-sum(j in Navf)x(i,j,k)
    ))
52     z(i,k) >= 0
53 end-do
54
55 forall(i in Nav, k in Ber)T(u,k)>=T(i,k)+t(i,k) - M*(1-sum(j in
    Navf)x(i,j,k))
56
57 forall(i in Navi, k in Ber) sum(j in Navf)x(i,j,k) <= 1
58
59 !Navio atendido em apenas 1 berco e apenas 1 vez
60 forall (i in Nav) sum(j in Navf, k in Ber) x(i,j,k) = 1
61 forall (j in Nav) sum(i in Navi, k in Ber) x(i,j,k) = 1
62 !forall (i in Nav) sum(k in Ber) x(i,i,k) = 0
63
64 !"Primeiro" navio

```

```

65 forall (k in Ber) do
66     sum(j in Navif) x(0,j,k) = 1
67 end-do
68 forall (k in Ber, j in Navif) do
69     x(j,0,k) = 0
70 end-do
71
72 !"Ultimo" navio
73 forall (k in Ber) do
74     sum(j in Navif) x(j,u,k) = 1
75 end-do
76 forall (k in Ber, j in Navif) do
77     x(u,j,k) = 0
78 end-do
79
80 !Continuidade de fluxo
81 forall (i in Nav, k in Ber) do
82     sum(j in Navf)x(i,j,k) - sum(j in Navi)x(j,i,k) = 0
83 end-do
84
85 !tempo atracacao maior que chegada
86 forall (i in Nav, k in Ber) do
87     T(i,k) >= Cheg(i) - M*(1-sum(j in Navf)x(i,j,k))
88 end-do
89
90 !Calculo do tempo de atracacao
91 forall (i in Nav, j in Nav, k in Ber) do
92     T(i,k) + t(i,k)*x(i,j,k) - T(j,k) <= (1-x(i,j,k))*M
93 end-do
94
95 !Atracacao no berco 2
96 forall(i in Nav,k in Nav) do
97     !tda(i,k) >= t(k,2)*y(i,k)
98     tda(i,k) <= M*y(i,k)
99     tdp(i,k) >= t(i,1)*(1-y(i,k))
100    tdp(i,k) <= M*(1-y(i,k))
101    T(k,2) + tda(i,k) = tdp(i,k) + T(i,1)
102 end-do
103
104
105 forall (i in Navif, j in Navif, k in Ber) x(i,j,k) is_binary
106 forall (i in Navf, k in Ber) T(i,k) >= 0
107 forall (i in Nav, j in Nav) y(i,j) is_binary

```

```

108 forall (i in Nav, j in Nav) tda(i,j) >= 0
109 forall (i in Nav, j in Nav) tdp(i,j) >= 0
110
111 minimize(obj)
112
113 end-model

```

A.2 Terminal dePraia Mole

```

1 model TPM
2 uses "mmxprs"; !gain access to the Xpress-Optimizer solver
3
4 !sample declarations section
5 declarations
6 ! Conjuntos
7     Ber: set of real
8     Nav: set of real
9     Navi: set of real
10    Navf: set of real
11    Navif: set of real
12    Des: set of string
13    EstA: set of string
14    EstB: set of string
15
16
17 ! Variaveis
18     x: array(Navif, Navif, Ber) of mpvar
19     T: array(Navif, Ber) of mpvar
20     t: array(Nav, Ber) of mpvar
21     DN: array(Navif, Navif, Des, Ber) of mpvar
22     !DNE: array(Nav, Des, EstA) of mpvar
23     TDN: array(Navif, Des) of mpvar
24     ES: array(Navif, Navif, EstA, Ber) of mpvar
25     TES: array(Navif, EstA) of mpvar
26     y: array(Nav, Ber) of mpvar
27     p: array(Nav, Des, Ber) of mpvar
28     q: array(Nav, EstA, Ber) of mpvar
29
30 ! Parametros
31     Cheg: array(Nav) of real
32     Carg: array(Nav) of real
33     M: real
34     VDes: array(Des) of real

```

```

35     VEst: array(EstA) of real
36     o: real
37     u: real
38     Aber: array(Ber) of real
39     !a: array(Des) of real
40     !b: array(EstA) of real
41     !w: real
42     !z: real
43 end-declarations
44
45 initializations from 'ValePMn-2.dat'
46     Ber as 'BERCOS'
47     Nav as 'NAVIOS'
48     Navi as 'NAVIOSi'
49     Navf as 'NAVIOSf'
50     Navif as 'NAVIOSif'
51     M as 'M'
52     Cheg as 'CHEGADA'
53     Carg as 'CARGA'
54     Des as 'DESCARREGADORES'
55     VDes as 'VASAO_DESCARREGADOR'
56     EstA as 'ESTEIRAS_A'
57     EstB as 'ESTEIRAS_B'
58     VEst as 'VASAO_ESTEIRA'
59     o as 'o'
60     u as 'u'
61     Aber as 'Aber'
62     !a as 'a'
63     !b as 'b'
64     !w as 'c'
65     !z as 'd'
66 end-initializations
67
68 obj:= sum(i in Nav, k in Ber)y(i,k)
69
70 forall(i in Nav, k in Ber) do
71     y(i,k) <= M*sum(j in Navf)x(i,j,k)
72     y(i,k) <= T(i,k)+0.5*t(i,k)-Cheg(i)
73     y(i,k) >= T(i,k)+0.5*t(i,k)-Cheg(i) - M*(1-sum(j in Navf)x(
74         i,j,k))
75     y(i,k) >= 0
76 end-do

```

```

77 !Tempo de servico
78 forall (i in Nav) sum(d in Des, k in Ber) p(i,d,k)*VDes(d) >= Carg(i
)
79 forall (i in Nav) sum(e in EstA, k in Ber) q(i,e,k)*VEst(e) >= Carg(
i)
80 forall(i in Nav, d in Des, k in Ber) do
81     p(i,d,k) <= M*(sum(j in Navf)DN(i,j,d,k))
82     p(i,d,k) <= t(i,k)
83     p(i,d,k) >= t(i,k) - M*(1 - sum(j in Navf)DN(i,j,d,k))
84     p(i,d,k) >= 0
85 end-do
86 forall(i in Nav, e in EstA, k in Ber) do
87     q(i,e,k) <= M*(sum(j in Navf)ES(i,j,e,k))
88     q(i,e,k) <= t(i,k)
89     q(i,e,k) >= t(i,k) - M*(1 - sum(j in Navf)ES(i,j,e,k))
90     q(i,e,k) >= 0
91 end-do
92
93 !Navio atendido em apenas 1 berco e apenas 1 vez
94 forall (i in Nav) sum(j in Navf, k in Ber) x(i,j,k) = 1
95 forall (j in Nav) sum(i in Navi, k in Ber) x(i,j,k) = 1
96
97 !"Primeiro" navio
98 forall (k in Ber) do
99     sum(j in Navif) x(o,j,k) = 1
100 end-do
101 forall (k in Ber, j in Navif) do
102     x(j,o,k) = 0
103 end-do
104
105 !"Ultimo" navio
106 forall (k in Ber) do
107     sum(j in Navif) x(j,u,k) = 1
108 end-do
109 forall (k in Ber, j in Navif) do
110     x(u,j,k) = 0
111 end-do
112
113 !Continuidade de fluxo
114 forall (i in Nav, k in Ber) do
115     sum(j in Navf)x(i,j,k) - sum(j in Navi)x(j,i,k) = 0
116 end-do
117

```

```

118 !tempo atracacao maior que chegada
119 forall (i in Nav, k in Ber) do
120     T(i,k) >= Cheg(i)! - M*(1-sum(j in Navf)x(i,j,k))
121 end-do
122 forall (i in Nav, k in Ber) do
123     T(i,k) >= Aber(k)! - M*(1-sum(j in Navf)x(i,j,k))
124 end-do
125
126 !Calculo do tempo de atracacao
127 forall (i in Nav, j in Nav, k in Ber) do
128     T(i,k) + t(i,k) - T(j,k) <= (1-x(i,j,k))*M
129 end-do
130
131 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

132
133 !Ordem dos descarregadores
134 forall (i in Nav) do
135     sum(j in Navif) DN(i,j,'DN07',1) >= sum(j in Navif) DN(i,j
136         , 'DN06',1)
137     sum(j in Navif) DN(i,j,'DN06',1) >= sum(j in Navif) DN(i,j
138         , 'DN05',1)
139     sum(j in Navif) DN(i,j,'DN05',1) >= sum(j in Navif) DN(i,j
140         , 'DN04',1)
141
142     sum(j in Navif) DN(i,j,'DN04',2) >= sum(j in Navif) DN(i,j
143         , 'DN05',2)
144     sum(j in Navif) DN(i,j,'DN05',2) >= sum(j in Navif) DN(i,j
145         , 'DN06',2)
146     sum(j in Navif) DN(i,j,'DN06',2) >= sum(j in Navif) DN(i,j
147         , 'DN07',2)
148 end-do
149
150
151 !Descarregadores navio inicial e final
152 forall (d in Des) do
153     sum(i in Navif, k in Ber) DN(o,i,d,k) = 1
154     sum(i in Navif, k in Ber) DN(i,u,d,k) = 1
155     forall (i in Navif, k in Ber) do
156         DN(i,o,d,k) = 0
157         DN(u,i,d,k) = 0
158         DN(i,i,d,k) = 0
159     end-do

```



```

154 end-do
155
156 !Descarregadores por navio
157 forall (i in Navi, k in Ber) do
158     sum(d in Des, j in Navf) DN(i,j,d,k) >= 1*(sum(j in Navf)x(
159         i,j,k))
160     sum(d in Des, j in Navf) DN(i,j,d,k) <= 3*(sum(j in Navf)x(
161         i,j,k))
162 end-do
163 forall (j in Navf, k in Ber) do
164     sum(d in Des, i in Navi) DN(i,j,d,k) >= 1*(sum(i in Navi)x(
165         i,j,k))
166     sum(d in Des, i in Navi) DN(i,j,d,k) <= 3*(sum(i in Navi)x(
167         i,j,k))
168 end-do
169
170 forall(k in Ber)T(o,k) = 0
171
172 !Continuidade de fluxo Descarregador
173 forall (i in Nav, d in Des) do
174     sum(j in Navf, k in Ber)DN(i,j,d,k) - sum(j in Navi, k in
175         Ber)DN(j,i,d,k) = 0
176 end-do
177
178 !Tempo descarregadores
179 forall (i in Nav, j in Navf, d in Des, k in Ber) TDN(i,d) + t(i,k)
180     - TDN(j,d) <= (1-DN(i,j,d,k))*M
181 forall (i in Nav, d in Des, k in Ber) TDN(i,d) <= T(i,k) + M*(1-sum
182     (j in Navf)DN(i,j,d,k))
183 forall (i in Nav, d in Des, k in Ber) TDN(i,d) >= T(i,k) - M*(1-sum
184     (j in Navf)DN(i,j,d,k))
185
186
187 !Esteiras navio inicial e final
188 forall (e in EstA) do
189     sum(i in Navif, k in Ber) ES(o,i,e,k) = 1
190     sum(i in Navif, k in Ber) ES(i,u,e,k) = 1
191     forall (i in Navif, k in Ber) do
192         ES(i,o,e,k) = 0
193         ES(u,i,e,k) = 0
194     end-do
195 end-do

```

```

188             ES(i,i,e,k) = 0
189         end-do
190 end-do
191
192 !Esteiras por navio
193 forall (i in Navi, k in Ber) do
194     sum(e in EstA, j in Navf) ES(i,j,e,k) >= 1*(sum(j in Navf)x
195         (i,j,k))
196     sum(e in EstA, j in Navf) ES(i,j,e,k) <= 2*(sum(j in Navf)x
197         (i,j,k))
198 end-do
199 forall (j in Navf, k in Ber) do
200     sum(e in EstA, i in Navi) ES(i,j,e,k) >= 1*(sum(i in Navi)x
201         (i,j,k))
202     sum(e in EstA, i in Navi) ES(i,j,e,k) <= 2*(sum(i in Navi)x
203         (i,j,k))
204 end-do
205
206 !Continuidade de fluxo Esteiras
207 forall (i in Nav, e in EstA) do
208     sum(j in Navf, k in Ber)ES(i,j,e,k) - sum(j in Navi, k in
209         Ber)ES(j,i,e,k) = 0
210 end-do
211
212 !Tempo Esteiras
213 forall (i in Nav, j in Navf, e in EstA, k in Ber) TES(i,e) + t(i,k)
214     - TES(j,e) <= (1-ES(i,j,e,k))*M
215 forall (i in Nav, e in EstA, k in Ber) TES(i,e) <= T(i,k) + M*(1-
216     sum(j in Navf)ES(i,j,e,k))
217 forall (i in Nav, e in EstA, k in Ber) TES(i,e) >= T(i,k) - M*(1-
218     sum(j in Navf)ES(i,j,e,k))
219
220 forall (i in Navif, j in Navif, k in Ber) x(i,j,k) is_binary
221 forall (i in Navf, k in Ber) T(i,k) >= 0
222 forall (i in Nav, k in Ber) t(i,k)>=0
223 forall (i in Navif, j in Navif, d in Des, e in EstA, k in Ber) do
224     DN(i,j,d,k) is_binary
225     TDN(i,d) >= 0
226     ES(i,j,e,k) is_binary
227     TES(i,e) >= 0
228 end-do

```

```
223 minimize(obj)
224
225 end-model
```