



Flávia Barbosa

O Problema de Alocação de Berços:
Aspectos Teóricos e Computacionais

Campinas
2014



UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E
COMPUTAÇÃO CIENTÍFICA

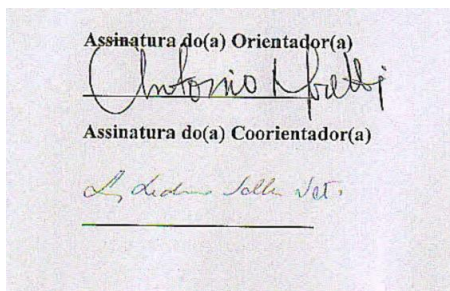
Flávia Barbosa

O Problema de Alocação de Berços:
Aspectos Teóricos e Computacionais

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 Mestra
em Matemática Aplicada

Orientador: Antônio Carlos Moretti
Coorientador: Luiz Leduino de Salles Neto

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA
DISSERTAÇÃO DEFENDIDA PELA ALUNA
FLÁVIA BARBOSA, E ORIENTADA PELO
PROF. DR. ANTÔNIO CARLOS MORETTI.



Campinas
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

B234p Barbosa, Flávia, 1989-
 O problema de alocação de berços : aspectos teóricos e computacionais /
 Flávia Barbosa. – Campinas, SP : [s.n.], 2014.

 Orientador: Antônio Carlos Moretti.

 Coorientador: Luiz Leduíno de Salles Neto.

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

 1. Programação linear. 2. Agenda de execução (Administração). 3. Heurística.
I. Moretti, Antônio Carlos, 1958-. II. Salles Neto, Luiz Leduíno de. III. Universidade
Estadual de Campinas. Instituto de Matemática, Estatística e Computação
Científica. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: The berth allocation problem : theoretical and computational aspects

Palavras-chave em inglês:

Linear programming

Scheduling (Management)

Heuristic

Área de concentração: Matemática Aplicada

Titulação: Mestra em Matemática Aplicada

Banca examinadora:

Antônio Carlos Moretti [Orientador]

Carla Taviane Lucke da Silva Ghidini

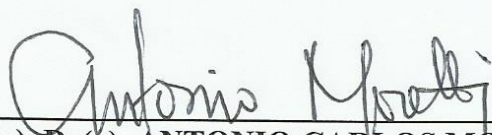
Anibal Tavares de Azevedo

Data de defesa: 20-01-2014

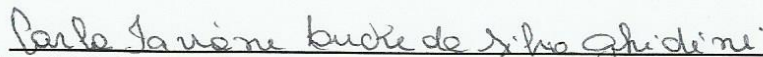
Programa de Pós-Graduação: Matemática Aplicada

Dissertação de Mestrado defendida em 20 de janeiro de 2014 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.



Prof.(a). Dr(a). ANTONIO CARLOS MORETTI



Prof.(a). Dr(a). CARLA TAVIANE LUCKE DA SILVA GHIDINI



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

Abstract

The international trading is highly subordinate on maritime transport and therefore, the ports have been forced to invest in groundwork and logistics. In this context, this work addresses the Berth Allocation Problem: how to allocate ships to berths in a given planning horizon so that the operational costs are minimized. In Brasil, the Vale company é responsible for the export of raw iron ore used in steel manufacturing. Thus, it will be proposed methods to optimize their port operations, more specifically to the Praia Mole Terminal at the Tubarão Port. For this purpose, two mathematical models and two heuristics were implemented. Mathematical models, adapted from literature cases are executed with GLPK and CPLEX and the results obtained are compared so that the best option is found.

Resumo

O comércio internacional é profundamente dependente do transporte marítimo e por isso, os portos tem sido forçados a investir em infra-estrutura e logística. Nesse contexto, o presente trabalho aborda o Problema de Alocação de Berços: como alocar navios a berços em um dado horizonte de planejamento de modo a minimizar os custos operacionais. No Brasil, a companhia Vale é responsável pela exportação da matéria-prima minério de ferro utilizado na fabricação do aço. Assim, será proposto métodos que otimizem suas operações portuárias, mais especificamente para o Terminal de Praia Mole no Porto de Tubarão. Para tanto, dois modelos matemáticos e duas heurísticas foram implementadas. Os modelos, adaptados de casos existentes na literatura, são executados com o GLPK e o CPLEX e os resultados obtidos são comparados para que a melhor opção seja encontrada.

Sumário

1	Introdução	1
2	Revisão Bibliográfica	4
2.1	Sobre a Companhia Vale	14
3	O Problema de Alocação de Berços	16
3.1	Um modelo geral para o problema de carregamento de <i>containers</i>	17
3.2	Redução do modelo para o caso bidimensional	20
3.3	Adaptação para Problema de Alocação de Berços	23
3.4	Modelo para o caso da Vale	25
3.5	Função-objetivo	26
3.5.1	Minimizar o maior tempo de saída dos navios	27
3.5.2	Minimizar a soma dos tempos de saída	28
3.5.3	Minimizar o maior tempo de espera	28
3.5.4	Minimizar a soma dos tempos de espera	29
3.6	Modelo de Esquematização de Navios a Berços	31
4	Heurística <i>Aloca-Berço</i>	32
4.1	Heurística <i>First Come, First Serve</i> (FCFS)	35
5	Experimentos Computacionais	38
5.1	Cenários com 10 navios	39
5.2	Cenários com 15 navios	42
5.3	Cenários com 20 navios	44
5.4	Cenários com 25 navios	46
5.5	Cenários com 30 navios	48
5.6	Cenários com 35 navios	50
5.7	Cenários com 40 navios	52

6	Conclusão	56
A	Dados gerados	61
B	Algoritmo da Heurística	74

Dedicatória

À minha avó Ignês,
por todo carinho em todos esses anos.
(In Memoriam)

Agradecimentos

Aos meus pais, Luiz e Márcia, pelo amor e dedicação em todo o tempo.

Ao meu irmão, Rodrigo, pelo companheirismo e apoio computacional.

Ao meu orientador, Prof. Dr. Moretti, pela ajuda e amizade.

Ao meu coorientador, Prof. Dr. Leduino, pela oportunidade de trabalhar no projeto.

A todos os amigos que, de uma maneira ou de outra, fizeram parte dessa jornada.

À FAPESP, pelo financiamento do projeto “Otimização dos custos de carregamento e de transporte de cargas por navios: aspectos teóricos e computacionais”.

Lista de Figuras

3.1	Interpretação do BAP como um problema de corte.	16
3.2	Problema tri-dimensional.	18
3.3	Problema bi-dimensional.	21
3.4	Problema de alocação da Vale.	26
3.5	Resultado do modelo que minimiza o maior tempo de saída. . .	28
3.6	Resultado do modelo que minimiza a soma tempos de saída. . .	29
3.7	Resultado do modelo que minimiza o maior tempo de espera. . .	30
3.8	Resultado do modelo que minimiza a soma dos tempos de espera. .	30
5.1	Comparações dos cenários com 10 navios.	41
5.2	Comparações dos cenários com 15 navios.	43
5.3	Comparações dos cenários com 20 navios.	45
5.4	Comparações dos cenários com 25 navios.	47
5.5	Comparações dos cenários com 30 navios.	49
5.6	Comparações dos cenários com 35 navios.	51
5.7	Comparações dos cenários com 40 navios.	54

Lista de Tabelas

5.1	Resultados para 10 navios.	40
5.2	Resultados para 15 navios.	42
5.3	Resultados para 20 navios.	44
5.4	Resultados para 25 navios.	46
5.5	Resultados para 30 navios.	48
5.6	Resultados para 35 navios.	50
5.7	Resultados para 40 navios.	53
A.1	Dados para 10 navios (I)	61
A.2	Dados para 10 navios (II)	62
A.3	Dados para 15 navios (I)	62
A.4	Dados para 15 navios (II)	63
A.5	Dados para 20 navios (I)	64
A.6	Dados para 20 navios (II)	65
A.7	Dados para 25 navios (I)	66
A.8	Dados para 25 navios (II)	67
A.9	Dados para 30 navios (I)	68
A.10	Dados para 30 navios (II)	69
A.11	Dados para 35 navios (I)	70
A.12	Dados para 35 navios (II)	71
A.13	Dados para 40 navios (I)	72
A.14	Dados para 40 navios (II)	73

Lista de Abreviaturas

BAP: Berth Allocation Problem
DBAP: Discrete Berth Allocation Problem
CBAP: Continuous Berth Allocation Problem
TBAP: Tactical Berth Allocation Problem
QCAP: Quay Crane Allocation Problem
B&CAP: Berth and Crane Allocation Problem
RPF: Relative Position Formulation
PAF: Position Assignment Formulation
CS: Clustering Search
GSSP: Generalized Set Partitioning Problem
GRASP: Greedy Randomized Adaptive Search Procedure
PR: Path Relinking
GA: Genetic Algorithm
FCFS: First-Come-First-Serve
MOGA: Multi Objective Genetic Algorithm

Capítulo 1

Introdução

O transporte marítimo é um dos pilares fundamentais do crescimento mundial da economia ao longo de toda a história. Atualmente o aumento da demanda do transporte em navios é um dos maiores problemas enfrentados por terminais marítimos ao redor do mundo. O intenso fluxo de navios e *containers* nos portos tem gerado uma busca por uma logística que minimize o tempo de espera e de atendimento dos navios, a quantidade de navios rejeitados, dentre outros. Além disso, um aumento da concorrência entre os portos marítimos, especialmente entre os geograficamente próximos, também ocorre como resultado desse crescimento.

Assim, surgiu uma maior dificuldade em gerenciar navios, tornando-se interessante analisar questões acerca do Problema de Alocação em Berços (Berth Allocation Problem - BAP) para navios em portos: como alocar navios a berços de modo a minimizar alguns desses objetivos. Decisões como onde e quando atracar os navios devem ser tomadas ao longo de um horizonte de planejamento levando em conta restrições de tempo e espaço, como o comprimento do navio, o horário de chegada, o número de *containers* para carregar ou descarregar e a localização do estoque da carga. As características do porto, da mercadoria, das demandas geram inúmeros problemas com pequenas, mas relevantes diferenças.

O problema reflete as principais decisões referentes aos recursos financeiros, afinal uma boa distribuição dos navios ao longo do cais proporcionará a satisfação dos proprietários dos navios e aumentará a produtividade do porto. Por isso, os fluxos contínuos de carga através de terminais requerem planos precisos; análises de decisão e técnicas de otimização devem ser aplicadas para melhorar várias operações.

Muitos dos problemas estudados na literatura são referente a portos que trabalham com a carga e a descarga de *containers*. A escala e a natureza dos problemas nesses terminais muitas vezes impossibilitam que as decisões tomadas sejam ótimas. A maioria dos berços de *containers* nos principais portos são concedidos a operadores de navios, a fim de que eles sejam responsáveis pelo processamento da carga e, assim, alcancem maior produtividade. Visto que isso se justifica no caso em que são negociados um grande volume de *containers* com um grande número de navios, talvez não seja possível garantir economias de custo. Os terminais então devem se preparar para atender o desafio de lidar com navios com capacidade muito elevada. Fatores-chave proporcionam a eficiência para o empilhamento e transporte dessa grande quantidade de carga. Alta produtividade e movimentação de *containers* a custos baixos são essenciais para a operação do terminal, e berços e guindastes são os recursos mais importantes para isso. Uma programação eficiente pode fazer uma operação bem sucedida, o que irá melhorar a satisfação do cliente, aumentar o fluxo de carga e levar a maiores rendimentos do porto.

Na maioria das vezes, o planejamento do cais e do guindaste são determinados separadamente, o que pode causar conflitos e levar a um planejamento inviável. Portanto, a alocação de berços e guindastes, que organiza as posições de atracação, o tempo de atracação e os guindastes para os navios que chegam, é uma nova tendência no ramo da pesquisa operacional. Uma alocação simultânea pode ajudar a reduzir muito os custos operacionais do terminal.

De acordo com [10], a companhia Vale é uma das maiores companhias mineradoras de ferro do mundo. A logística de movimentação dos produtos representa ponto primordial em seus processos, e tem tido especial atenção pelas equipes que buscam otimizar e aperfeiçoar o desempenho da empresa.

Em [29] o Porto de Tubarão é apresentado contendo seis regiões. Cada uma delas trabalha com um tipo específico de produto. O píer 1 é dividido em lado sul e lado norte. O lado sul possui um comprimento de cais acostável de 340,3 metros e o lado norte possui um comprimento acostável de 353,7 metros. Ambos são responsáveis pela manipulação do minério de ferro. O píer 2 também trabalha com o minério de ferro, possuindo um comprimento acostável de 365 metros. O píer 3 é destinado ao carregamento de grãos a granel e possui um cais acostável máximo de 280 metros. O píer 4 é destinado à movimentação de produtos diversos, como fertilizantes, *containers*, pallets e carga geral. Seu comprimento de cais acostável é de 230 metros. O terminal de granéis líquidos pertence a Vale, mas é operado pela Petróleo Brasileiro

S.A - Petrobrás e BR Distribuidora, para movimentação de derivados de uso. Por fim, há o Terminal de Praia Mole, que opera descarregamento de carvão metalúrgico e outros granéis sólidos compatíveis. Possui um cais acostável de 705,67 metros e pode operar dois navios. C

Sendo assim, a proposta aqui é apresentar o melhor modelo para a movimentação do Terminal de Praia Mole, considerando a condição de dois berços em operação. Uma vez que o problema em questão é referente à carga granel, os modelos encontrados na literatura para terminais de *containers* serão readequados para o contexto encontrado. No Capítulo 2, é apresentada uma breve revisão bibliográfica sobre as variações do Problema de Alocação de Berços que podem ser encontradas na literatura, incluindo o objeto de estudo: a companhia Vale. No Capítulo 3, dois modelos matemáticos são propostos para resolver o Problema de Alocação de Berços. No Capítulo 4, a heurística “*Aloca-Berço*” foi construída com o intuito de resolver o problema em um tempo mais eficaz e a heurística “*First Come First Serve*” foi implementada. No Capítulo 5, os resultados dos dois modelos e das duas heurísticas são comparados e analisados para diferentes cenários gerados. No Capítulo 6, é apresentada uma conclusão sobre o problema estudado.

Capítulo 2

Revisão Bibliográfica

Vários modelos para o BAP já foram propostos na literatura. As diferenças dizem respeito a algumas suposições que são assumidas, como, por exemplo, se existe a possibilidade de espera de navios, se vários navios podem atracar no mesmo berço, se o tempo de chegada dos navios é considerado, se o tempo de atendimento é proporcional ao tamanho do navio, entre outras. Baseado em problemas que ocorreram no porto de Hong Kong, em [12] é considerado um problema de alocação de espaços em berços para navios, permitindo vários navios sendo atracados por berço e considerando os horários de chegadas dos navios (problema dinâmico), com o objetivo de minimizar o tempo total de escoamento (a soma dos tempos de espera e de serviço de um navio) ponderado (os pesos refletem a importância relativa dos navios). Como nesse porto o espaço dos berços é limitado, surgiu a necessidade da criação de uma alocação de berços eficiente para a administração do tráfego de *containers*. Um berço típico nos terminais pode acomodar vários navios ao mesmo tempo, sendo que, quando não há espaço disponível no berço para um navio atracar, ele deve esperar.

Na maioria dos problemas, o objetivo é minimizar os custos referentes ao porto e ao navio. Isso envolve os custos de espera, as multas por atraso e a utilização de equipamentos como guindastes. Em algumas variações do problema também podem ser incluídas outras condições de penalidade na função objetivo, como prioridade de atendimento, valor de carga movimentada, e outras.

A administração em vários terminais de *containers* tem se esforçado para reduzir os custos através de uma eficiente utilização de recursos (mão-de-obra, berços, guindastes, equipamentos). Os berços são os recursos mais

importantes e uma boa alocação acarreta uma melhora na satisfação dos clientes e um aumento no rendimento do porto. Sendo assim, em [17] é proposto um modelo que busca maximizar a utilização do cais satisfazendo algumas restrições para atracação de navios de *containers* usando uma abordagem analítica. Isso é feito através da minimização dos custos de penalidade resultantes de atrasos nas partidas dos navios e dos custos adicionais de atendimento resultantes de localizações desfavoráveis dos navios no cais. Para esses pesquisadores, a posição da atracação é uma variável de decisão muito importante: se um navio é colocado em uma posição perto do estoque dos *containers* que devem ser carregados, então o custo de transferência dos *containers* por carregadores pode ser diminuído.

Alguns portos na Europa e na China, em que há um tráfego intenso de navios e *containers*, são chamados de *multi-users terminals*: terminais com um longo cais, onde uma grande quantidade de navios chegam e são simultânea e dinamicamente alocados, nem sempre na mesma posição. Esse tipo de terminal, de acordo com [15], é um sistema generalizado muito usado para portos nessas condições, uma vez que a produtividade depende de uma alocação eficiente dos navios nos berços. Assim, o terminal é particionado em vários berços e a alocação dos navios no local apropriado é baseado nas características desses berços, como comprimento e profundidade. O problema associado é chamado de Problema de Alocação em Berços Discreto (Discrete Berth Allocation Problem - DBAP). No entanto, o artigo é fundamentado no fato de que navios algumas vezes são ancorados através da fronteira do berço, ou seja, a atracação dos navios é feita em um espaço contínuo (Continuous Berth Allocation Problem - CBAP).

Em [2], o problema é classificado em algumas categorias. O processo de chegada dos navios pode ser considerado estático ou dinâmico. No primeiro caso, o horário de chegada dos navios não impõe uma restrição sobre os horários de atracação: os navios podem atracar a qualquer momento dada que uma porção do cais esteja disponível. No segundo caso, os navios não podem atracar antes de chegarem ao porto. Quanto à chegada dos navios, ela podem ser determinística, quando valores esperados de horários de chegadas são dados, ou estocástica, em que uma distribuição de chegadas é dada com incertezas acerca dos horários. O tempo de atendimento também pode ter classificações diferentes: fixo e invariável, dependente da posição onde o navio foi alocado ou ainda estocástico (interrupções podem acontecer trazendo incertezas nos tempos de serviço). Em relação às restrições espaciais, além das duas propostas por [15], aqui é considerado o caso híbrido, no qual o cais é

particionado em um conjunto de berços mas um navio pode ocupar mais de um berço e mais de um navio pode ocupar um dado berço.

Há também outra maneira de dividir o espaço físico do terminal. No problema estudado em [9], o terminal é dividido em vários cais. Cada cais consiste em um conjunto de vários berços, que por sua vez são divididos em seções. Cada cais corresponde a um estiramento linear de espaço. Cada navio pode fisicamente ser ancorado através de diferentes seções, mas não pode ser ancorado através de diferentes cais. Nesse artigo, o objetivo é alocar um espaço de atracação para os navios em tempo real, garantindo que quase todos os navios sejam atracados assim que chegam ao terminal e estejam perto das localizações de preferência. Tal alocação deve ser determinada dinamicamente sobre o tempo.

Na literatura, é possível encontrar algumas variações do BAP. Uma delas é chamada em [11] de Problema Tático de Alocação de Berços (Tactical Berth Allocation Problem - TBAP). A diferença entre esse problema e o BAP está na duração do horizonte de planejamento: o planejamento no TBAP consiste em meses, permitindo a integração entre os custos do terminal de um modo mais compreensivo. Em um terminal de transbordo, os *containers* chegam e partem por navios sendo temporariamente estocados no pátio. Quando um navio é descarregado, os *containers* devem ser alocados em posições no pátio perto o suficiente do berço onde está atracado o navio em que serão carregados, acelerando o serviço desse navio. Entretanto, quando a posição de embarque de um *container* é longe do seu pátio, o *container* deve ser realocado antes da chegada do navio. Consequentemente, a administração do pátio trata de uma alocação dinâmica de *containers*. Vale notar que o TBAP determina a longo termo qual é o melhor berço de atracação para cada navio, induzindo o fluxo de *containers* dentro do pátio. Assim, os custos de pátio são um efeito da atribuição simultânea dos navios a seus berços de origem. Outra diferença diz respeito ao horários de chegada dos navios. No TBAP as linhas de transporte indicam o intervalo de tempo para o horário de chegada esperado. O planejamento tático de berços deve acomodar tais horários de chegada ou uma combinação alternativa deve ser encontrada. Por isso, o problema aqui é saber se acomodar uma demanda de um cliente é factível e quais são os impactos sobre o terminal. Além disso, uma vez que o tempo de serviço depende de fatores como quantidade de *containers* para serem carregados ou descarregados, distribuição desses dentro do navio e número de guindastes atribuídos para o navio, o TBAP trata com a negociação entre o terminal e as linhas de transporte sobre uma

atribuição reservada de guindastes ao longo dos turnos de trabalho.

Os guindastes são os equipamentos fundamentais no terminal usados para a movimentação de *containers*. O emprego ineficiente desses pode ser o gargalo para operações rápidas nos navios. Em decorrência disso, surge o problema de alocação de guindastes (Quay Crane Allocation Problem - QCAP): depois que um navio chega ao terminal, guindastes descarregam os *containers* do navio, que são levados até os pátios de estocagem. O mesmo acontece com navios que devem ser carregados para exportação. No QCAP assume-se que um navio já está alocado em uma posição do berço e espera para ser atendido pelos guindastes. Para portos com grande movimentação, é muito caro que o navio permaneça esperando por muito tempo na sua posição de atracação. Por isso, é importante que tantos guindastes quanto necessários sejam usados, maximizando o número de *containers* carregados ou descarregados por unidade de tempo. Entretanto, esses equipamentos são caros e certamente não são um investimento de custo eficiente. Por consequência, um terminal competitivo precisa maximizar a eficiência das operações sobre um número limitado de guindastes.

Segundo [28], existem duas regras que restringem a operação dos guindastes. A primeira é a restrição de *non-crossing*: os guindastes devem ser mantidos em uma sequência fixa, pois dividem um trilho comum ao longo do berço. A segunda é uma restrição de separação mínima: os guindastes precisam manter uma distância mínima entre eles por questões de segurança. Aqui é estudado o problema de alocação de guindastes dinâmico: um número fixo de guindastes em um berço com comprimento limitado atende uma série de navios que chegam sequencialmente no berço com o passar do tempo. O objetivo é maximizar a quantidade de *containers* manipulados por unidade de tempo sujeito às duas restrições já citadas. Esse problema apresenta complicações consequentes de condições climáticas, *containers* perdidos e congestionamento de tráfego no terminal. Por esse motivo, a maioria dos estudos encontrados na literatura resolvem o problema em uma configuração estática, em que um dado número de navios no berço são pré-atribuídos a um conjunto de guindastes durante o horizonte de planejamento. O problema é resolvido como um problema de agendamento de máquinas: se qualquer parâmetro no sistema muda, então o problema precisa ser resolvido. Porém, o caso estático não é ótimo e é difícil de ser implementado para um longo período de tempo.

Um navio grande pode ser simultaneamente processado por vários guindastes. Normalmente, o tempo de atendimento de um navio diminui quando

mais guindastes estão atribuídos a ele. Analogamente, o tempo de atendimento de um navio será maior do que o esperado se um número insuficiente de guindastes estiverem operando, atrasando a partida do navio. Infelizmente, não é possível atribuir muitos guindastes a um mesmo navio: o equipamento é caro e os terminais não possuem uma grande quantidade, o que torna um custo de oportunidade em uma perda real se há um guindaste ocioso. Além disso, muitos guindastes em um mesmo navio pode causar interferência em vez de aprimoramento. Em [20], algumas suposições são feitas na tentativa de simplificar a complexidade do problema modelado. Em primeiro lugar, o QCAP sempre aproveita o resultado do BAP: considera-se que o problema de alocação de berços já foi resolvido e as posições de atracação, os tempos de chegada e partida e as cargas de trabalho dos navios são dados de entrada para o problema de alocação de guindastes. Depois, cada navio é dividido longitudinalmente em baias, normalmente do mesmo tamanho: cada baia acomoda uma fila de pilhas de *containers* (assim o tamanho do navio é dado em função do número de baias). O comprimento total do berço e a distância de segurança entre guindastes adjacentes também são dados em função do número de baias. Ou seja, todas as distâncias espaciais podem ser medidas em função do número de baias, e por isso uma baia será utilizada como uma unidade de comprimento. Em seguida, os guindastes são considerados idênticos, tanto em termos de produtividade quanto em termos de velocidade de movimentação e, uma vez que um guindaste inicia o atendimento de uma baia, ele só pára depois que terminar de atender toda a carga de trabalho dessa. Por último, o tempo total de preparação para o navio partir e esperar é assumido constante para todos os navios. A carga de trabalho de uma baia é medida em termos do tempo levado para um guindaste processar todos os *containers* da baia. Então, o objetivo é minimizar o tempo de processamento do navio.

Em geral, BAP e QCAP vêm associados entre si, originando um problema de alocação simultâneo (B&CAP), como é apresentado em [13]. O B&CAP consiste em dois subproblemas: BAP e QCAP. Para o primeiro subproblema, é considerado o mesmo modelo do DBAP utilizadas em [14]. Para obter a formulação B&CAP, as seguintes restrições do QCAP são adicionadas: guindastes se movem de um berço para o outro em um cais sobre um trilho e a transferência de guindastes não pode passar por berços em que existam navios sendo atendidos. Nesse caso, o movimento dos guindastes são tratados em mais detalhes, e para isso novas variáveis de decisão precisam ser definidas. O objetivo é minimizar o tempo total de serviço.

Os objetivos nos estudos anteriores do problema simultâneo são quase sempre os mesmos: minimizar o tempo de estadia (ou de espera) dos navios. Mas em [27] é ressaltado que, apesar da possibilidade do aumento da satisfação das companhias de navio e da competitividade entre os portos, isso pode acarretar efeitos negativos, como por exemplo um custo de produção alto. Um planejamento eficaz deve considerar ambos. Levando em conta essa consideração, é estabelecido um modelo que minimiza o tempo de retorno médio dos navios e os custos de produção ao mesmo tempo.

Cada um dos modelos acima é favorecido por um método de resolução diferente. Algoritmos genéticos são usados com frequência devido a seu potencial como técnica de otimização singular.

O modelo de alocação de berços pode ser representado em um diagrama espaço-tempo em que o eixo horizontal representa o horizonte de planejamento e o eixo vertical representa o comprimento do berço. Logo, o navio é interpretado como um retângulo cujo comprimento é o tempo do processamento e a altura é o tamanho do navio. Esse modelo foi formulado de duas maneiras em [12]. A primeira delas é chamada de *Formulação da Posição Relativa - RPF*, que considera a posição relativa dos retângulos no diagrama. Para esse caso, existe uma solução ótima em que cada navio (retângulo) j está imediatamente a direita de outro navio ou é o primeiro navio e imediatamente acima de outro navio ou está alocado na primeira seção. A outra é a chamada *Formulação da Posição de Designação - PAF*, em que é considerado o espaço coberto pelos retângulos. O diagrama é particionado em blocos, cada um deles com a altura igual ao tamanho de uma seção e largura igual a uma unidade de tempo. É proposto uma Relaxação Lagrangeana, que fornece um limitante inferior para o PAF. Para obter um limitante melhor, é usado o método do subgradiente. Então, é descrito um procedimento de árvore de busca que utiliza as propriedades de ambas formulações, no qual os retângulos são adicionados no diagrama espaço-tempo um a um. A árvore de busca é ramificada da seguinte maneira. Os k retângulos inseridos formam uma escada em cada nó da árvore. Os degraus da escada indicam posições factíveis potenciais para inserção de outros retângulos. Quando um retângulo é inserido em uma etapa, seu tempo de atribuição é o valor máximo entre hora de início da etapa e o tempo de chegada do navio. Para reduzir a ramificação, limitantes inferiores e superiores são usados para gerar simplificações do processo. Nesse artigo também é proposta uma heurística que agrupa os navios em lotes de acordo com seus horários de chegada, a qual gera uma solução inicial. Em seguida, os navios são permutados em dois lotes

consecutivos se a permutação reduz o valor da função objetivo. Só então o procedimento exato da árvore de busca é aplicado para cada lote resultante.

Essa representação também é utilizada em [15], porém com os eixos invertidos (o comprimento do retângulo representa o comprimento do navio e a altura do retângulo representa a duração do atendimento). A borda inferior do retângulo representa o horário do início do atendimento e a superior o horário do final. O espaço no cais pode ser representado geometricamente por retângulos infinitamente longos onde seus comprimentos são o comprimento do cais e a altura define o tempo. Quando dois navios são alocados lado a lado uma certa distância de folga deve existir entre eles, a qual será incluída no comprimento do navio. Além disso, também é observado um tempo de folga quando um navio é alocado em um berço onde um navio foi anteriormente atendido, a qual será incluída no tempo de serviço de cada navio. O BAP contínuo (CBAP) estudado aqui consiste em determinar as coordenadas dos retângulos ao longo do cais de modo que todos os retângulos representando navios não sejam sobrepostos nem alocados abaixo de suas linhas horizontais pré-determinadas, que representam o horário de chegada dos navios. O limitante superior para o BAPC é obtido da solução ótima do BAP discreto (DBAP) quando o comprimento do berço é determinado pelo comprimento máximo do navio envolvido no problema. Já o limitante inferior é obtido quando o comprimento do berço é determinado por um valor muito pequeno.

Em [17] as soluções do BAP com a mesma representação são codificadas por uma sequência de navios para o uso de um algoritmo de *simulated annealing*. Para decodificar a sequência de navios em uma solução, deve ser possível obter posições próximas do ótimo dos retângulos de navios no diagrama tempo-espaço para uma dada sequência de navios. Um x -cluster é definido como um conjunto de retângulos cujos lados verticais estão em contato uns com os outros, e um y -cluster é definido como um conjunto de retângulos cujos lados horizontais estão em contato uns com os outros. Um cluster de navios será considerado estável se o custo total para os navios nele não pode ser reduzido movimentando-se o cluster na direção positiva ou negativa. Quando um x -cluster é estável, ou um ou mais retângulos estão localizados nos seus pontos de custo mínimo no eixo x ou um lado de pelo menos um retângulo do cluster está na fronteira esquerda ou direita do cais. Quando um y -cluster é estável, os horários de atracação de todos os navios ou estão localizados no intervalo compreendendo a [horário de chegada, tempo de operação - horário de partida] ou são seus horários de chegadas ou zero.

Seguindo a mesma interpretação, em [9] os planos de atracação são codificados por pares de permutação (H, V) de todos os navios. Um grafo direcionado é associado às restrições envolvendo as variáveis de decisão de tempo e um problema de *custo-tempo* é formulado. Um outro grafo é associado às restrições envolvendo as variáveis de decisão de espaço e então é formulado um problema de *custo-espaço*. A abordagem sugere um método efetivo para obter um bom empacotamento dado um par de sequência fixo que é buscado no espaço com um algoritmo de *Simulated Annealing*.

O *Clustering Search* (CS) apresentado em [23] é formado basicamente por três componentes principais: uma metaheurística geradora de soluções, um processo de agrupamento e uma heurística de busca local. A cada iteração, uma solução S é gerada pela metaheurística e enviada para o processo de agrupamento a qual é então agrupada no cluster mais similar C_j e o centro desse cluster c_i é atualizado com informações da nova solução agrupada, fazendo com que o centro se desloque no espaço de busca. Em seguida, é analisado o volume v_j do cluster e, se esse volume atingir um limitante λ , significa que algum padrão de solução está sendo predominantemente gerado e esse cluster pode estar em uma região de busca promissora. Por fim, é analisado o índice de ineficácia r_j : caso a heurística de busca local não melhore a solução, é aplicada uma perturbação aleatória no centro para evitar um possível ótimo local. Caso contrário, a heurística de busca local é aplicada no centro analisando a vizinhança do cluster. Encerrado esse processo, retorna-se para a metaheurística que irá gerar uma nova solução.

Em [3] e [26] o BAP foi modelado como um problema de partição de conjuntos generalizado (Generalized Set Partitioning Problem - GSSP). Assumindo que todas as medições de tempo são consideradas inteiras, uma variável (coluna) representa uma atribuição factível de um único navio a um berço em um horário específico. As primeiras n linhas correspondem aos n navios. Se uma coluna representa uma atribuição no navio i , então haverá o valor 1 na linha i e zeros no restante das n primeiras linhas. Além disso, existe uma linha para cada unidade de tempo disponível em cada berço. Uma entrada em uma coluna é igual a 1 se o navio ocupa o berço na unidade de tempo considerada, caso contrário é zero. O custo de cada coluna é igual ao tempo de serviço decorrentes da atribuição navio/berço/horário.

O modelo inteiro misto foi proposto em [4] como uma extensão do problema de gerenciamento de máquinas para o problema de planejamento integrado. O objetivo é minimizar o *makespan* (horário de término da sequência das tarefas) melhorando a ligação entre os equipamentos e aumentando a

produção e assim foi resolvido por um algoritmo baseado em *busca tabu*. Tal método permite movimentos afastando-se de uma solução ótima que torna a função objetivo pior na tentativa de obter uma solução melhor. Para tanto, é sugerido uma nova estrutura de vizinhança para tratar o problema.

Uma outra abordagem, em [21] é baseada na aplicação do método *Greedy Randomized Adaptive Search Procedure* (GRASP) para construção de soluções de forma integrada com o método *Path Relinking* (PR) como uma estratégia de intensificação de busca para resolução do PAB. Cada iteração do GRASP é composta por uma fase construtiva, na qual uma solução viável é construída, e por uma fase de busca local que, baseada na solução construída, aplica movimentos iterativos até que o ótimo local seja encontrado. Em seguida, o PR é aplicado como uma estratégia de intensificação, explorando as trajetórias que conectam a solução da iteração atual à melhor solução obtida até então.

Em [25] foi proposto um algoritmo genético (GA) simplificado que possa analisar uma lista de navios (com horário de chegada, tamanho do calado¹, comprimento e tipo de carga) e uma lista de berços (com os horários de disponibilidade, profundidade e comprimento) para em seguida verificar se as restrições impostas são satisfeitas e, posteriormente, alocar um berço a um navio. O algoritmo tem como critério para alocação verificar o último momento de liberação de um dado berço, somar o período de intervalo a ser considerado e alocar um navio a esse berço. Assim, a sequência de navios é representada por um cromossomo e uma população inicial é formada: os indivíduos são codificados em uma sequência de navios, e cada componente da sequência é um gene (navio). A avaliação de performance representa a capacidade do indivíduo adaptar-se ao meio, e essa medida deve ser considerada de modo que, quanto menor seu valor, maior sua capacidade de adaptação. Inicialmente os indivíduos são ordenados em ordem decrescente de desempenho (o primeiro representa o melhor desempenho). O cruzamento é feito considerando a média das posições ocupadas pelos indivíduos nos cromossomos e, em seguida, são ordenados em uma lista em ordem crescente. A mutação é realizada utilizando-se uma probabilidade de ocorrência.

Em [18] é sugerida uma abordagem híbrida para o QCAP, que combina GA com uma heurística. Sabe-se que utilizando-se somente GA, é possível encontrar regiões promissoras para um ótimo global, mas algumas vezes há uma certa dificuldade para encontrar o mínimo exato. A motivação em usar

¹Profundidade a que se encontra o ponto mais baixo de uma embarcação.

uma abordagem híbrida é a possibilidade de obter sistemas com desempenhos melhores que exploram e unem vantagens de cada uma das estratégias puras. Como as soluções encontradas pelo GA ainda podem ser melhoradas, a proposta da hibridização é facilitar a convergência local. O procedimento utilizado pode ser dividido em quatro fases. Na primeira fase é criada uma sequência de navios: a relação de precedência dos navios é decidida, uma prioridade aleatória é gerada para cada um e a sequência é determinada pela regra *First-Come-First-Served* (FCFS). Na segunda, os navios são alocados a berços: através da geração de um número aleatório para cada navio. Na terceira, os guindastes são atribuídos aos berços com a geração de um número aleatório para cada berço. Na quarta fase ocorre a designação dos berços e o planejamento dos guindastes: os guindastes são transferidos entre os berços satisfazendo a designação e é feito um diagrama de Draw Gantt para o planejamento. Assim sendo, o procedimento do *Cross-Over* é realizado por um método de um ponto de corte (selecionado aleatoriamente) para cada navio na sequência e uma estratégia de mapeamento é combinada para garantir a factibilidade da nova descendência. Para o procedimento da mutação, foi adaptada uma mutação de permutação para gerar a nova descendência.

Uma variação do GA também foi usada em [27] para resolver o problema de alocação de berços e guindastes. Foi sugerido um método de solução de três fases. A primeira fase consiste em um algoritmo genético multi-objetivo (MOGA) para gerar ou iterar soluções factíveis que podem definir a ordem de serviço, as posições de atracação e o número de guindastes para cada navio. A segunda fase gera o horário de atracação por uma heurística no planejamento de guindastes. Por último é selecionado um planejamento final por soluções de Pareto².

O problema de alocação simultânea B&CAP de [13] foi resolvido utilizando-se uma heurística baseada em GA para estimar uma solução aproximada. A cada iteração (geração), haverá a execução de dois procedimentos: um é o procedimento do algoritmo genético - crossover e mutações - para a alocação da ordem dos navios nos berços sem considerar a programação dos guindastes e o outro é a heurística para a programação dos guindastes dada uma ordem de alocação dos navios nos berços alternativa definida por um cromossomo. O algoritmo genético é caracterizado pela transformação do valor da função objetivo em um valor de capacidade. Como o cromossomo define apenas a

²Uma solução de Pareto é ótima se não for possível melhorar a situação sem piorar a situação de outro agente.

ordem de alocação dos navios nos berços, antes do processo de reprodução a programação dos guindastes é executada de modo que o valor da função objetivo é produzido para ser utilizado para o valor de capacidade. Já na reprodução, cada cromossomo individual é copiado de acordo com a função de capacidade pela estratégia de torneio. Para o crossover, é utilizado o crossover de dois pontos. Já para a fase da programação dos guindastes, a factibilidade da programação do atendimento dos navios é verificada de acordo com o número de guindastes disponíveis e as tarefas dos guindastes são reprogramadas. O cromossomo determina a ordem de designação dos navios aos berços. Se existem vários guindastes, os navios são atendidos assim que chegam ao porto e os berços designados se tornam ociosos após o serviço. Porém, se não existem guindastes suficientes, alguns navios podem ter o atendimento atrasado. Por isso, é necessário examinar se o número de guindastes dado ao sistema como parâmetro de entrada é grande o suficiente para os navios serem servidos em uma programação inicial. Então, o atraso do atendimento é minimizado por uma programação ótima das tarefas dos guindastes.

2.1 Sobre a Companhia Vale

Segundo [1], a empresa Vale tem como foco de negócio a extração e exportação da matéria-prima minério de ferro, mineral ferroso utilizado na fabricação do aço. Uma vez que seus clientes localizam-se a uma longa distância das minas e o consumo de carvão no Brasil aumentou, surgiu a necessidade à Vale adaptar seu processo logístico de operação, incluindo o pátio ferroviário de Tubarão, onde os vagões são carregados e os trens formados para iniciar a viagem ao longo da Estrada de Ferro Vitória-Minas. Por isso, tornou-se fundamental identificar as características dos subsistemas que o compõem e analisar as suas inter-relações. O sistema logístico de transporte do carvão inicia no berço do Terminal de Praia Mole e termina nos clientes situados no Estado de Minas Gerais, contemplando descarga de navio, correias transportadoras, pátios de estocagem, silo de carregamento, pátio ferroviário de manobra e formação de trens, transporte ferroviário ao longo da Estrada de Ferro Vitória-Minas, pátio de recepção, manobra e desmembramento de trens e descarga nos clientes. Nesse estudo, o sistema logístico é dividido em quatro subsistemas. O primeiro deles é o porto, que compreende o berço do Terminal de Praia Mole onde estão posicionados os descarregadores de navio

e cujo cais permite a atracação simultânea de dois navios, as correias transportadoras e o Pátio de estocagem com recuperadoras, empilhadeiras e pás mecânicas. O segundo e mais complexo é o pátio ferroviário, onde são feitas as operações de separação, consolidação, classificação, manutenção, carga e descarga de vagões de minério de ferro e carga geral. O terceiro compreende a linha ferroviária representada pela Estrada de Ferro Vitória-Minas, a qual liga o Porto de Tubarão às minas de minério de ferro, às siderúrgicas e possui conexões com a Ferrovia Centro-Atlântica. O quarto e último é o cliente, pátios de destino e entrega dos vagões às siderúrgicas que os descarregam.

Para que o produto da Vale seja competitivo e atraente no comércio mundial apesar das grandes distâncias do mercado consumidor, o controle de toda a cadeia logística para escoar o minério de ferro deve ser aprimorado, desde a sua origem, nas minas, até os países mais distantes. Como descrito em [8], no píer de Praia Mole ocorre o desembarque de carvão em dois berços de atracação através de quatro descarregadores de navios de caçamba, com capacidade aproximada de 7000 toneladas cada. Após esta operação, o carvão é então levado por transportadores de correia até o pátio de estocagem, onde será empilhado até ser vendido.

Visando o estudo de aspectos teóricos e computacionais da otimização de carregamento e do transporte de cargas por navios, em especial os enfrentados pela empresa Vale S.A., serão desenvolvidos dois modelos e métodos heurísticos de otimização para esse problema, levando em conta os horários de chegada dos navios e o tempo necessário para processá-los. Assim, uma solução ótima deverá ser fornecida ou pelo menos um conjunto de soluções eficientes que possa auxiliar a tomada de decisão da empresa.

Capítulo 3

O Problema de Alocação de Berços

O Problema de Alocação de Berços (Berth Allocation Problem - BAP) consiste em determinar os horários e as posições de atracação de navios nos berços do cais. Primeiramente será trabalhada uma interpretação de problema de corte como é mostrado na Figura 3.1.

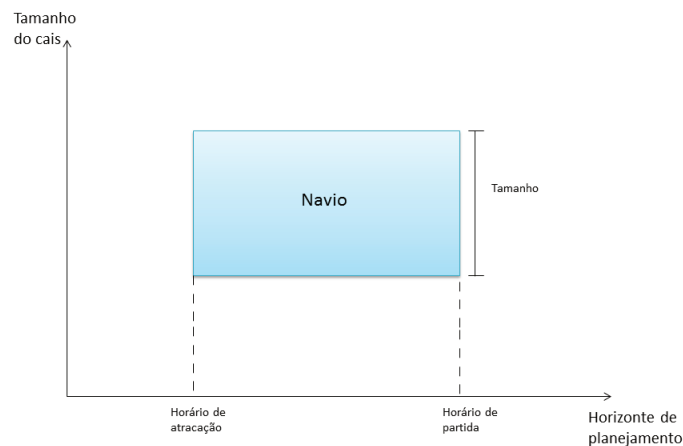


Figura 3.1: Interpretação do BAP como um problema de corte.

No caso que será estudado aqui, o eixo horizontal representa o horizonte

de planejamento e o eixo vertical representa o tamanho do cais. Analogamente, o navio será representado como um retângulo cujo comprimento é o tempo de processamento do navio e cuja altura é tamanho do navio. Assim, nosso problema será determinar as coordenadas do canto inferior esquerdo do retângulo no diagrama.

Para tanto, a partir de um modelo de problema de corte tridimensional, descrito em [6] como problema de carregamento de *containers* será construído o problema de alocação de berços.

3.1 Um modelo geral para o problema de carregamento de *containers*

O problema em questão é selecionar um número de *containers* para empacotar um dado conjunto de caixas, ambos com tamanhos variados. O objetivo primário é minimizar o espaço inutilizado, e para isso será considerado um modelo matemático que irá obter uma solução ótima com garantia. Será assumido que as dimensões de cada *container* são conhecidas e que as dimensões e quantidades de caixas são dadas. Cada caixa é tratada de forma independente e deve ser colocada ortogonalmente em um *container* como pode ser observado na Figura 3.2 (o ponto em destaque representa as coordenadas que devem ser determinadas):

Sem perda de generalidade, suponha que a maior dimensão da caixa é seu comprimento, a menor é sua altura e a intermediária é sua largura (ou seja, as caixas podem ser rotacionadas). Os parâmetros e as variáveis usadas serão definidas a seguir:

- N : número total de caixas;
- m : número de *containers* disponíveis;
- M : número grande arbitrário;
- (p_i, q_i, r_i) : parâmetros indicando respectivamente o comprimento, a largura e a altura da caixa i ;
- (L_j, W_j, H_j) : parâmetros indicando respectivamente o comprimento, a largura e a altura do *container* j ;

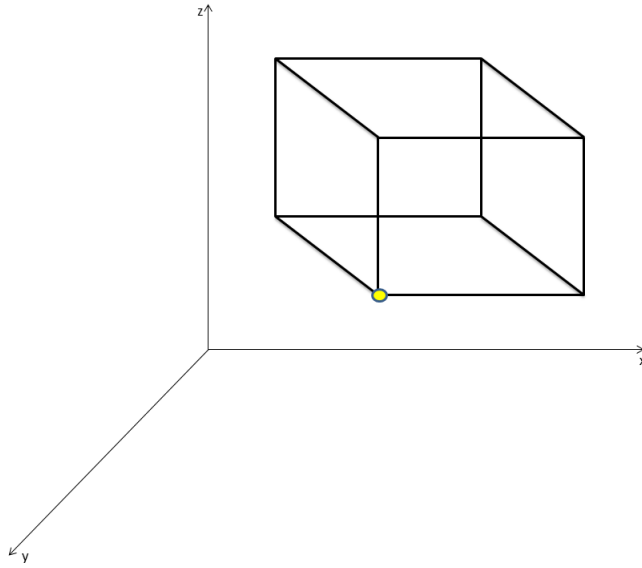


Figura 3.2: Problema tri-dimensional.

- (x_i, y_i, z_i) : variáveis contínuas indicando as coordenadas do canto inferior esquerdo da caixa i .

$$s_{ij} = \begin{cases} 1 & \text{se a caixa } i \text{ é colocada no } \textit{container } j. \\ 0 & \text{caso contrário.} \end{cases}$$

$$n_j = \begin{cases} 1 & \text{se o } \textit{container } j \text{ é utilizado.} \\ 0 & \text{caso contrário.} \end{cases}$$

$$l_{ti} = \begin{cases} 1 & \text{se o comprimento da caixa } i \text{ é paralelo ao eixo } t. \\ 0 & \text{caso contrário.} \end{cases}$$

$$w_{ti} = \begin{cases} 1 & \text{se a largura da caixa } i \text{ é paralela ao eixo } t. \\ 0 & \text{caso contrário.} \end{cases}$$

$$h_{ti} = \begin{cases} 1 & \text{se a altura da caixa } i \text{ é paralela ao eixo } t. \\ 0 & \text{caso contrário.} \end{cases}$$

$$a_{ik} = \begin{cases} 1 & \text{se a caixa } i \text{ está à esquerda da caixa } k. \\ 0 & \text{caso contrário.} \end{cases}$$

$$b_{ik} = \begin{cases} 1 & \text{se a caixa } i \text{ está à direita da caixa } k. \\ 0 & \text{caso contrário.} \end{cases}$$

$$c_{ik} = \begin{cases} 1 & \text{se a caixa } i \text{ está atrás da caixa } k. \\ 0 & \text{caso contrário.} \end{cases}$$

$$d_{ik} = \begin{cases} 1 & \text{se a caixa } i \text{ está na frente da caixa } k. \\ 0 & \text{caso contrário.} \end{cases}$$

$$e_{ik} = \begin{cases} 1 & \text{se a caixa } i \text{ está abaixo da caixa } k. \\ 0 & \text{caso contrário.} \end{cases}$$

$$f_{ik} = \begin{cases} 1 & \text{se a caixa } i \text{ está acima da caixa } k. \\ 0 & \text{caso contrário.} \end{cases}$$

Sendo assim, o problema é então formulado:

$$\min \quad \sum_{j=1}^m L_j W_j H_j n_j - \sum_{i=1}^N p_i q_i r_i \quad (3.1)$$

$$\text{s.a} \quad x_i + p_i l_{xi} + q_i w_{xi} + r_i h_{xi} \leq x_k + (1 - a_{ik})M \quad \forall i, k, i < k \quad (3.2)$$

$$x_k + p_k l_{xk} + q_k w_{xk} + r_k h_{xk} \leq x_i + (1 - b_{ik})M \quad \forall i, k, i < k \quad (3.3)$$

$$y_i + q_i w_{yi} + p_i l_{yi} + r_i h_{yi} \leq y_k + (1 - c_{ik})M \quad \forall i, k, i < k \quad (3.4)$$

$$y_k + q_k w_{yk} + p_k l_{yk} + r_k h_{yk} \leq y_i + (1 - d_{ik})M \quad \forall i, k, i < k \quad (3.5)$$

$$z_i + r_i h_{zi} + q_i w_{zi} + p_i l_{zi} \leq z_k + (1 - e_{ik})M \quad \forall i, k, i < k \quad (3.6)$$

$$z_k + r_k h_{zk} + q_k w_{zk} + p_k l_{zk} \leq z_i + (1 - f_{ik})M \quad \forall i, k, i < k \quad (3.7)$$

$$a_{ik} + b_{ik} + c_{ik} + d_{ik} + e_{ik} + f_{ik} \geq s_{ij} + s_{ik} - 1 \quad \forall i, k, i < k \quad (3.8)$$

$$\sum_{j=1}^m s_{ij} = 1 \quad \forall i \quad (3.9)$$

$$\sum_{i=1}^N s_{ij} \leq M n_j \quad \forall j \quad (3.10)$$

$$x_i + p_i l_{xi} + q_i w_{xi} + r_i h_{xi} \leq L_j + (1 - s_{ij})M \quad \forall i, j \quad (3.11)$$

$$y_i + q_i w_{yi} + p_i l_{yi} + r_i h_{yi} \leq W_j + (1 - s_{ij})M \quad \forall i, j \quad (3.12)$$

$$z_i + r_i h_{zi} + q_i w_{zi} + p_i l_{zi} \leq H_j + (1 - s_{ij})M \quad \forall i, j \quad (3.13)$$

$$l_{ti}, w_{ti}, h_{ti}, a_{ik}, b_{ik}, c_{ik}, d_{ik}, e_{ik}, f_{ik}, s_{ij}, n_j \in \{0, 1\} \quad \forall i, k, j, t \quad (3.14)$$

$$x_i, y_i, z_i \geq 0 \quad \forall i. \quad (3.15)$$

A Função-objetivo (3.1) minimiza o espaço total não utilizado dos *containers* selecionados. As Restrições (3.2) e (3.3) garantem que não haverá sobreposição de caixas na direção x , a (3.4) e a (3.5) na direção y e a (3.6) e a (3.7) na direção z . A Restrição (3.8) garante que a verificação de não sobreposição feita pelas seis restrições anteriores ocorra somente para pares de caixas colocadas no mesmo *container*, a Restrição (3.9) não permite que uma caixa seja colocada em mais de um container, e a Restrição (3.10) força que o *container* seja considerado como utilizado se qualquer caixa é alocada nele. As Restrições (3.11), (3.12) e (3.13) verificam se as caixas colocadas no *container* respeitam as dimensões do mesmo. As duas últimas restrições especificam a natureza das variáveis utilizadas.

3.2 Redução do modelo para o caso bidimensional

Partindo do problema apresentado na seção anterior, será obtido um modelo para um problema de corte bidimensional: como alocar retângulos em uma folha de estoque também retangular sem que haja sobreposição. Para tanto, sem perda de generalidade, o eixo z não será considerado, bem como as informações referentes à dimensão intermediária das caixas (largura) (Figura 3.3).

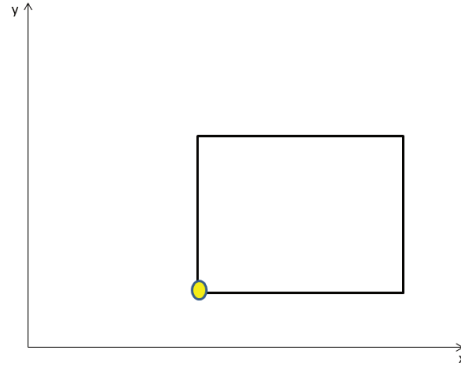


Figura 3.3: Problema bi-dimensional.

Para isso, somente as seguintes informações e variáveis são necessárias:

- N : número de retângulos a serem alocados;
- m : número total de folhas de estoque disponíveis;
- M : número grande arbitrário;
- (p_i, r_i) : parâmetros indicando respectivamente o comprimento e a altura da caixa i ;
- (L_j, H_j) : parâmetros indicando respectivamente o comprimento e a altura do *container* j ;
- (x_i, y_i) : variáveis contínuas indicando as coordenadas do canto inferior esquerdo da caixa i .

$$s_{ij} = \begin{cases} 1 & \text{se o retângulo } i \text{ é colocado na folha de estoque } j. \\ 0 & \text{caso contrário.} \end{cases}$$

$$n_j = \begin{cases} 1 & \text{se a folha de estoque } j \text{ é utilizada.} \\ 0 & \text{caso contrário.} \end{cases}$$

$$l_{ti} = \begin{cases} 1 & \text{se o comprimento do retângulo } i \text{ é paralelo ao eixo } t. \\ 0 & \text{caso contrário.} \end{cases}$$

$$h_{ti} = \begin{cases} 1 & \text{se a altura do retângulo } i \text{ é paralela ao eixo } t. \\ 0 & \text{caso contrário.} \end{cases}$$

$$a_{ik} = \begin{cases} 1 & \text{se o retângulo } i \text{ está à esquerda do retângulo } k. \\ 0 & \text{caso contrário.} \end{cases}$$

$$b_{ik} = \begin{cases} 1 & \text{se o retângulo } i \text{ está à direita do retângulo } k. \\ 0 & \text{caso contrário.} \end{cases}$$

$$c_{ik} = \begin{cases} 1 & \text{se o retângulo } i \text{ está atrás do retângulo } k. \\ 0 & \text{caso contrário.} \end{cases}$$

$$d_{ik} = \begin{cases} 1 & \text{se o retângulo } i \text{ está na frente do retângulo } k. \\ 0 & \text{caso contrário.} \end{cases}$$

O seguinte modelo é obtido:

$$\min \quad \sum_{j=1}^m L_j H_j n_j - \sum_{i=1}^N p_i r_i \quad (3.16)$$

$$\text{s.a} \quad x_i + p_i l_{xi} + r_i h_{xi} \leq x_k + (1 - a_{ik})M \quad \forall i, k, i < k \quad (3.17)$$

$$x_k + p_k l_{xk} + r_k h_{xk} \leq x_i + (1 - b_{ik})M \quad \forall i, k, i < k \quad (3.18)$$

$$y_i + p_i l_{yi} + r_i h_{yi} \leq y_k + (1 - c_{ik})M \quad \forall i, k, i < k \quad (3.19)$$

$$y_k + p_k l_{yk} + r_k h_{yk} \leq y_i + (1 - d_{ik})M \quad \forall i, k, i < k \quad (3.20)$$

$$a_{ik} + b_{ik} + c_{ik} + d_{ik} \geq s_{ij} + s_{ik} - 1 \quad \forall i, k, i < k \quad (3.21)$$

$$\sum_{j=1}^m s_{ij} = 1 \quad \forall i \quad (3.22)$$

$$\sum_{i=1}^N s_{ij} \leq Mn_j \quad \forall j \quad (3.23)$$

$$x_i + p_i l_{xi} + r_i h_{xi} \leq L_j + (1 - s_{ij})M \quad \forall i, j \quad (3.24)$$

$$y_i + p_i l_{yi} + r_i h_{yi} \leq W_j + (1 - s_{ij})M \quad \forall i, j \quad (3.25)$$

$$l_{ti}, h_{ti}, a_{ik}, b_{ik}, c_{ik}, d_{ik}, s_{ij}, n_j \in \{0, 1\} \quad \forall i, k, j, t \quad (3.26)$$

$$x_i, y_i \geq 0 \quad \forall i. \quad (3.27)$$

A Função-objetivo (3.16) minimiza o espaço total não utilizado das folhas de estoque selecionadas. As Restrições (3.17) e (3.18) garantem que não haverá sobreposição de retângulos na direção x , a (3.19) e a (3.20) na direção y . A Restrição (3.21) garante que a verificação de não sobreposição feita pelas seis restrições anteriores ocorra somente para pares de retângulos colocadas na mesma folha de estoque, a Restrição (3.22) não permite que um retângulo seja colocada em mais de uma folha de estoque, e a Restrição (3.23) força que a folha de estoque seja considerado como utilizado se qualquer caixa é alocada nele. As Restrições (3.24) e (3.25) verificam se os retângulos colocados na folha de estoque respeitam as dimensões da mesma. As duas últimas restrições especificam a natureza das variáveis utilizadas.

3.3 Adaptação para Problema de Alocação de Berços

No problema que está sendo trabalhado, algumas variáveis terão seus valores fixados por motivos de coerência a partir do modelo bidimensional apresentado, afinal as rotações não fazem sentido.

Aqui, a folha retangular de estoque representará um diagrama tempo-espaço; portanto será fixa, de modo que sua menor dimensão seja a altura do diagrama, representando o tamanho do cais e sua maior dimensão seja o comprimento, representando o horizonte de planejamento. Já os retângulos representarão os navios, e por isso já terão sua orientação definida: a altura será o tamanho e o comprimento será o tempo de processamento. Assim:

$$l_{xi} = 1 \text{ e } h_{xi} = 0;$$

ou seja, o comprimento do retângulo é paralelo ao eixo x e

$$l_{yi} = 0 \text{ e } h_{yi} = 1;$$

isso é, a altura do retângulo é paralela ao eixo y .

Existe apenas um diagrama, $m = 1$, e por isso, a função-objetivo deverá ser alterada já que minimizar a quantidade de espaço total inutilizado não é coerente. Mais adiante serão analisados 4 tipos de função-objetivo e a mais interessante ao problema será escolhida. O horizonte de planejamento em questão é ilimitado, e por esse motivo, a restrição $x_i + p_i l_{xi} + r_i h_{xi} \leq L$ é eliminada do problema. Pelo mesmo motivo, as restrições (3.22) e (3.23) são eliminadas do problema.

Além disso, uma restrição que garanta que os navios serão alocados somente após sua chegada precisará ser adicionada. Seja e_i o parâmetro que denota essa informação para o navio i . Então, a seguinte restrição é construída:

$$x_i \geq e_i \quad \forall i.$$

Para que o modelo esteja completo, há a necessidade de limitar o horário máximo de saída dos navios, parâmetro que será denotado por s_i . Assim, restrição a seguir é adicionada ao problema:

$$x_i + p_i \leq s_i \quad \forall i.$$

Eliminando as variáveis que dizem respeito à quantidade de retângulos usados e fixando as variáveis l_{ti} e h_{ti} do modo como foi explicado, obtem-se a formulação:

$$\min \quad \textit{objetivo} \quad (3.28)$$

$$\text{s.a} \quad x_i + p_i \leq x_k + (1 - a_{ik})M \quad \forall i, k, i < k \quad (3.29)$$

$$x_k + p_k \leq x_i + (1 - b_{ik})M \quad \forall i, k, i < k \quad (3.30)$$

$$y_i + r_i \leq y_k + (1 - c_{ik})M \quad \forall i, k, i < k \quad (3.31)$$

$$y_k + r_k \leq y_i + (1 - d_{ik})M \quad \forall i, k, i < k \quad (3.32)$$

$$a_{ik} + b_{ik} + c_{ik} + d_{ik} \geq 1 \quad \forall i, k, i < k \quad (3.33)$$

$$y_i + r_i \leq W \quad \forall i \quad (3.34)$$

$$x_i \geq e_i \quad \forall i \quad (3.35)$$

$$x_i + p_i \leq s_i \quad \forall i \quad (3.36)$$

$$l_{xi}, h_{yi}, a_{ik}, b_{ik}, c_{ik}, d_{ik} \in 0, 1 \quad \forall i, k \quad (3.37)$$

$$x_i, y_i \geq 0 \quad \forall i. \quad (3.38)$$

em que p_i é o tamanho no navio i e r_i é o tempo de processamento do navio i . O “objetivo” será definido adiante.

3.4 Modelo para o caso da Vale

Para esse caso, mais algumas alterações devem ser feitas.

Como o cais tem apenas dois berços de mesmo tamanho e cada berço só acomoda um navio por vez, a variável y_i será canalizada e terá um novo significado:

$$1 \leq y_i \leq 2$$

ou seja, se $y_i = 1$, então o navio i está no berço 1, e se $y_i = 2$, então o navio i está no berço 2. Sendo assim, assumamos $r_i = 1 \quad \forall i$, já que cada navio ocupa apenas um berço, como mostra a Figura 3.4:

Além disso, não será considerado um limite de horário de partida para os navios.

Assim, o problema é formulado da seguinte maneira:

$$\min \quad \textit{objetivo} \quad (3.39)$$

$$\text{s.a} \quad x_i + p_i \leq x_k + (1 - a_{ik})M \quad \forall i, k, i < k \quad (3.40)$$

$$x_k + p_k \leq x_i + (1 - b_{ik})M \quad \forall i, k, i < k \quad (3.41)$$

$$y_i + 1 \leq y_k + (1 - c_{ik})M \quad \forall i, k, i < k \quad (3.42)$$

$$y_k + 1 \leq y_i + (1 - d_{ik})M \quad \forall i, k, i < k \quad (3.43)$$

$$a_{ik} + b_{ik} + c_{ik} + d_{ik} \geq 1 \quad \forall i, k, i < k \quad (3.44)$$

$$x_i \geq e_i \quad \forall i \quad (3.45)$$

$$1 \leq y_i \leq 2 \quad (3.46)$$

$$a_{ik}, b_{ik}, c_{ik}, d_{ik} \in 0, 1 \quad \forall i, k \quad (3.47)$$

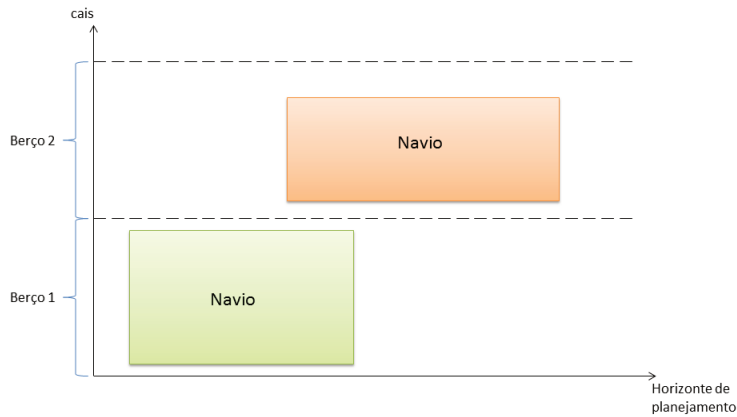


Figura 3.4: Problema de alocação da Vale.

$$x_i \geq 0 \quad \forall i \quad (3.48)$$

$$y_i \geq 0 \text{ e inteiro } \forall i. \quad (3.49)$$

3.5 Função-objetivo

Para que a função-objetivo que melhor se adapta ao problema real seja escolhida, serão consideradas 4 diferentes e seus resultados serão analisados. Os dados utilizados para tal análise foram retirados de [17]:

navio i	p_i	e_i	s_i
1	7	12	200
2	22	22	460
3	13	27	530
4	6	28	360
5	14	11	340
6	7	21	320
7	18	5	300

(os tempos de saída do artigo não foram utilizados)

3.5.1 Minimizar o maior tempo de saída dos navios

A primeira função-objetivo analisada é minimizar o maior tempo de saída dos navios. Para cada navio i , o horário que deixará o berço é dado por $x_i + p_i$. Assim temos a função:

$$\min \max_i \{x_i + p_i\}. \quad (3.50)$$

Como não é interessante trabalhar com um problema não-linear, a seguinte linearização é feita:

$$\min \quad u \quad (3.51)$$

$$\text{s.a} \quad u \geq x_i + p_i \quad \forall i. \quad (3.52)$$

O resultado para o problema de alocação foi:

navio i	x_i	y_i
1	23	2
2	30	2
3	33	1
4	46	1
5	12	1
6	26	1
7	5	2

Na Figura 3.5 pode-se ver como ficou essa alocação, com um ótimo de 52. A soma dos tempos de espera nesse caso foi 49.

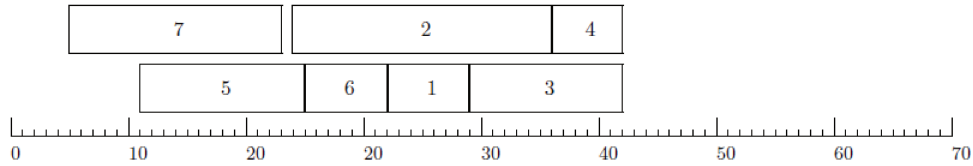


Figura 3.5: Resultado do modelo que minimiza o maior tempo de saída.

3.5.2 Minimizar a soma dos tempos de saída

Ainda trabalhando com o horário de partida do berço, considere uma função-objetivo que minimiza a soma dos tempos de saída de todos os navios que passam pelo porto:

$$\min \sum_i (x_i + p_i) \quad (3.53)$$

Os resultados para o modelo foi:

navio i	x_i	berço
1	12	1
2	33	1
3	36	2
4	30	2
5	19	1
6	23	2
7	5	2

Com valor ótimo de 245. A soma dos tempos de espera para essa configuração foi 32.

Na Figura 3.6 encontra-se um esquema dessa alocação.

3.5.3 Minimizar o maior tempo de espera

Seja $(e_i - x_i)$ o tempo que um navio fica esperando para começar a ser atendido. Considere uma função que minimiza o maior tempo de espera:

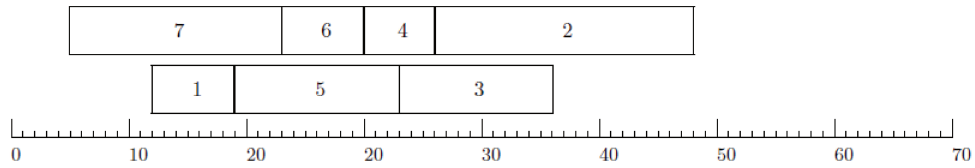


Figura 3.6: Resultado do modelo que minimiza a soma tempos de saída.

$$\min \max_i \{e_i - x_i\} \quad (3.54)$$

A seguinte linearização é feita:

$$\min \quad u \quad (3.55)$$

$$\text{s.a} \quad u \geq e_i - x_i \quad \forall i. \quad (3.56)$$

Essa função dá o seguinte resultado:

navio i	x_i	berço
1	12	2
2	33	2
3	38	1
4	30	1
5	19	2
6	23	1
7	5	1

O ótimo atinge o valor 11 e a soma dos tempos de espera foi 34.

3.5.4 Minimizar a soma dos tempos de espera

Analisando novamente o tempo de espera dos navios, considere uma função-objetivo que minimiza a soma de todos os tempos de espera:

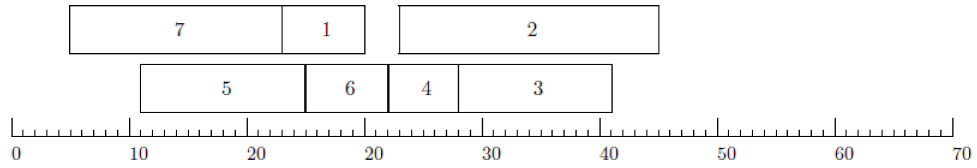


Figura 3.7: Resultado do modelo que minimiza o maior tempo de espera.

$$\min \sum_i (e_i - x_i) \quad (3.57)$$

O resultado obtido para o problema foi:

navio i	x_i	berço
1	12	1
2	33	1
3	36	2
4	30	2
5	19	1
6	23	2
7	5	2

Com valor ótimo igual a 32.

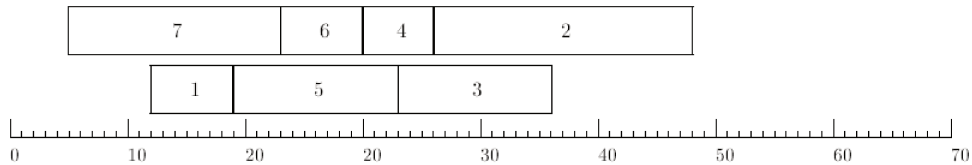


Figura 3.8: Resultado do modelo que minimiza a soma dos tempos de espera.

3.6 Modelo de Esquematização de Navios a Berços

Além do modelo de problema de corte, foi proposto também um modelo de esquematização em máquinas para resolver o problema de alocação de berços. Os parâmetros aqui são os mesmos utilizados anteriormente; as variáveis utilizadas serão as seguintes:

$$z_{ij} = \begin{cases} 1 & \text{se o navio } i \text{ está no berço } j. \\ 0 & \text{caso contrário.} \end{cases}$$

$$a_{ik} = \begin{cases} 1 & \text{se o retângulo } i \text{ está à esquerda do retângulo } k. \\ 0 & \text{caso contrário.} \end{cases}$$

x_i = posição do início do atendimento do navio no horizonte de planejamento.

Assim, o seguinte problema é modelado:

$$\min \quad \text{objetivo} \quad (3.58)$$

$$\text{s.a} \quad \sum_j z_{ij} = 1 \quad \forall i \quad (3.59)$$

$$z_{ij} + z_{kj} - a_{ik} - a_{ki} \leq 1 \quad \forall i \neq k, j \quad (3.60)$$

$$x_i + p_i - (1 - a_{ik})M \leq x_k \quad \forall i, k \quad (3.61)$$

$$z_{ij} + z_{kh} + a_{ik} + a_{ki} \leq 2 \quad \forall i \neq k, j \neq h \quad (3.62)$$

$$x_i \geq e_i \quad \forall i \quad (3.63)$$

$$x_i \geq 0 \quad \forall i. \quad (3.64)$$

$$a_{i,k}, z_{i,j} \in \{0, 1\} \quad \forall i, k, j \quad (3.65)$$

A Restrição (3.59) garante que cada navio será alocado em apenas um berço. A Restrição (3.60) mostra que se os navios i e k estão no mesmo berço, então ou o navio i está à direita do navio k ou o contrário ocorre. A Restrição (3.61) não permite que os navios estejam sobrepostos sobre o horizonte de planejamento. A (3.62) reforça que a variável $a_{i,k}$ só existe para navios que estão no mesmo berço. Por fim, a Restrição (3.63) não permite que os navios sejam atracados antes de sua chegada ao cais. As duas últimas especificam o tipo de variáveis utilizadas.

Para a função-objetivo deve ser feita a mesma análise do problema de corte.

Capítulo 4

Heurística *Aloca-Berço*

Além dos dois modelos apresentados no Capítulo 3, também será proposta uma heurística como uma opção mais rápida para resolver o problema de alocação de berços. Os parâmetros usados pela heurística são os mesmos: e_i é o horário de chegada e p_i o tempo de processamento do navio i . A primeira ação a ser feita é colocar os navios em ordem crescente de horário de partida ($w_i = e_i + p_i$). Então são criadas duas listas vazias, uma representando o Berço 1 e outra o Berço 2. Também deve ser criado um vetor L em que na posição j é armazenada a soma do tempo de atendimento do Berço j ($j = 1, 2$).

No Passo k do algoritmo (quando o k -ésimo navio deve ser alocado a um berço) verifica-se qual berço possui o menor tempo total de atendimento. Se $L(1) < L(2)$, então o Navio k é alocado no Berço 1 e $L(1)$ é atualizado para $L(1) + p(k)$. Caso contrário, o Navio k é alocado no Berço 2 e $L(2)$ é atualizado para $L(2) + p(k)$. Esse procedimento é feito até que todos os navios sejam atribuídos a um berço. Assim, ao final poderá ser calculado a soma dos tempos de espera.

dados os vetores p e e
 calcule $w = p + e$ e coloque em ordem crescente
 inicialize $L = [0, 0]$, Berço 1 = $\{\}$ e Berço 2 = $\{\}$
 para cada navio k em w ordenado
 — se $L(1) < L(2)$
 — então Berço 1 = Berço 1 \cup $\{\text{navio}(k)\}$;
 — $L(1) = L(1) + p(\text{navio}(k))$
 — caso contrário
 — Berço 2 = Berço 2 \cup $\{\text{navio}(k)\}$;
 — $L(2) = L(2) + p(\text{navio}(k))$
 — fim
 fim

Exemplo: Considere os dados do exemplo no Capítulo 3. Calculando $w_i = e_i + p_i$:

$$\mathbf{w} = \begin{pmatrix} 19 \\ 44 \\ 40 \\ 34 \\ 25 \\ 28 \\ 23 \end{pmatrix}$$

e ordenando, obtém-se o seguinte vetor:

$$\bar{\mathbf{w}} = \begin{pmatrix} 19 & (\text{navio1}) \\ 23 & (\text{navio7}) \\ 25 & (\text{navio5}) \\ 28 & (\text{navio6}) \\ 34 & (\text{navio4}) \\ 40 & (\text{navio3}) \\ 44 & (\text{navio2}) \end{pmatrix}$$

No início, Berço 1 = $\{\}$, Berço 2 = $\{\}$ e $L = [0, 0]$. O primeiro navio é alocado no berço 2. Daí:

$$\text{Berço 1} = \{\}$$

$$\begin{aligned} \text{Berço 2} &= \{1\} \\ L &= [0, 7] \end{aligned}$$

Em seguida, como $L(1) < L(2)$, o navio 7 é alocado no berço 1 e:

$$\begin{aligned} \text{Berço 1} &= \{7\} \\ \text{Berço 2} &= \{1\} \\ L &= [18, 7] \end{aligned}$$

A cada passo do algoritmo tem-se:

$$\begin{aligned} \text{Berço 1} &= \{7\} \\ \text{Berço 2} &= \{1, 5\} \\ L &= [18, 21] \end{aligned}$$

$$\begin{aligned} \text{Berço 1} &= \{7, 6\} \\ \text{Berço 2} &= \{1, 5\} \\ L &= [25, 21] \end{aligned}$$

$$\begin{aligned} \text{Berço 1} &= \{7, 6\} \\ \text{Berço 2} &= \{1, 5, 4\} \\ L &= [25, 27] \end{aligned}$$

E no final o resultado é dado por:

$$\begin{aligned} \text{Berço 1} &= \{7, 6, 3\} \\ \text{Berço 2} &= \{1, 5, 4\} \\ L &= [38, 27] \end{aligned}$$

E no final o resultado é dado por:

$$\begin{aligned}
\text{Berço 1} &= \{7, 6, 3\} \\
\text{Berço 2} &= \{1, 5, 4, 2\} \\
L &= [38, 49]
\end{aligned}$$

e a soma dos tempos de espera dos 7 navios foi 35.

4.1 Heurística *First Come, First Serve* (FCFS)

O porto que está sendo trabalhado aqui utiliza o princípio *First Come, First Serve* para a atribuição de seus navios nos dois berços: o primeiro navio a chegar é o primeiro a ser alocado ao berço que estiver livre. É interessante implementar esse algoritmo para que seja possível comparar os resultados dos modelos e da heurística com a estratégia de alocação que tem sido usado pela Vale.

Para tanto, basta fazer uma simples modificação no algoritmo da heurística apresentada anteriormente: dados e_i e p_i , os navios devem ser colocados em ordem crescente de horário de chegada. Seja $v = [e \ p]$.

Exemplo:

$$\mathbf{v} = \begin{pmatrix} 12 & 7 \\ 22 & 22 \\ 27 & 13 \\ 28 & 6 \\ 11 & 14 \\ 21 & 7 \\ 5 & 18 \end{pmatrix}$$

Ordenado v tem-se:

$$\bar{\mathbf{v}} = \begin{pmatrix} 5 & 18 & (\text{navio7}) \\ 11 & 14 & (\text{navio5}) \\ 12 & 7 & (\text{navio1}) \\ 21 & 7 & (\text{navio6}) \\ 22 & 22 & (\text{navio2}) \\ 27 & 13 & (\text{navio3}) \\ 28 & 6 & (\text{navio4}) \end{pmatrix}$$

Assim, o cenário com 7 navios utilizado no início do capítulo apresenta o seguinte resultado:

$$\begin{aligned}\text{Berço 1} &= \{\} \\ \text{Berço 2} &= \{7\} \\ L &= [0, 18]\end{aligned}$$

$$\begin{aligned}\text{Berço 1} &= \{5\} \\ \text{Berço 2} &= \{7\} \\ L &= [14, 18]\end{aligned}$$

$$\begin{aligned}\text{Berço 1} &= \{5, 1\} \\ \text{Berço 2} &= \{7\} \\ L &= [21, 18]\end{aligned}$$

$$\begin{aligned}\text{Berço 1} &= \{5, 1\} \\ \text{Berço 2} &= \{7, 6\} \\ L &= [21, 25]\end{aligned}$$

$$\begin{aligned}\text{Berço 1} &= \{5, 1, 2\} \\ \text{Berço 2} &= \{7, 6\} \\ L &= [43, 25]\end{aligned}$$

$$\begin{aligned}\text{Berço 1} &= \{5, 1, 2\} \\ \text{Berço 2} &= \{7, 6, 3\} \\ L &= [43, 38]\end{aligned}$$

$$\begin{aligned}\text{Berço 1} &= \{5, 1, 2\} \\ \text{Berço 2} &= \{7, 6, 3, 4\} \\ L &= [43, 44]\end{aligned}$$

e a soma dos tempos de espera dos 7 navios foi 43. Pode-se notar que o resultado foi maior que o tempo encontrado pela Heurística *Aloca-Berço*.

Capítulo 5

Experimentos Computacionais

Com base nos resultados obtidos para as diferentes funções-objetivo, nota-se que o mais importante economicamente é minimizar a soma dos tempos de espera, pois alguns portos pagam multa caso o atendimento de um navio demore muito a ser efetuado. Sendo assim, serão estudados como os dois modelos e as duas heurísticas fazem a alocação dos navios para diferentes cenários.

Problemas de diferentes tamanhos serão considerados: 10, 15, 20, 25, 30, 35 e 40 navios, e para cada um desses tamanhos serão gerados 10 cenários. Os horários de chegada dos navios foram gerados com base em uma distribuição uniforme entre $[0, 25]$ e os tempos de processamento foram gerados por uma distribuição binomial com 16 tentativas e $p = 0.5$, ambos gerados no Matlab.

A distribuição uniforme é uma distribuição de probabilidades contínua em que a probabilidade de se gerar qualquer ponto em um intervalo contido no espaço amostral é proporcional ao tamanho do intervalo $[\alpha, \beta]$. Sua função densidade de probabilidade é dada por:

$$f(x) = \begin{cases} \frac{1}{\beta - \alpha} & \text{para } \alpha \leq x \leq \beta \\ 0 & \text{caso contrário.} \end{cases}$$

A distribuição binomial é uma distribuição de probabilidade discreta do número de sucessos em uma sequência de n experimentos independentes do tipo sim/não, cada um deles com uma probabilidade de sucesso p . Sua função probabilidade é:

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}.$$

Os valores encontrados para a função-objetivo (soma dos tempos de espera) com GLPK e com CPLEX (versão 12.5) serão comparados para ambos modelos, o valor da função objetivo do problema de corte e do problema de esquematização quando utilizamos o CPLEX. Em cada cenário foi escolhido o modelo cuja função-objetivo retornou o melhor valor, o qual foi comparado com as heurísticas. O critério de parada para a execução dos problemas nos solvers foi o tempo, limitado em 1 hora. O computador utilizado para a execução foi um Dell Inspiron 14Z com processador Intel Core I5-3337U 1.80GHz, memória RAM de 6GB e um **HD** de 500 GB.

5.1 Cenários com 10 navios

Para esses cenários, o modelo de problema de corte obteve 266 restrições e 200 variáveis, das quais 190 são inteiras. Já o problema de esquematização de máquinas obteve 491 restrições e 130 variáveis, das quais 120 são inteiras.

Na Tabela 5.1 estão os valores ótimos obtidos para 10 cenários com 10 navios. Aqui o valor ótimo foi atingido em todos os cenários quando o problema de esquematização é executado no GLPK e no CPLEX e o problema de corte no CPLEX. Para o problema de corte no GLPK, apenas o cenário 7 atingiu o ótimo antes de 1h.

cenário	corte GLPK	scheduling GLPK	corte CPLEX	scheduling CPLEX	<i>“Aloca-Berço”</i>	FCFS
1	83	82	82	82	92	96
2	51	51	51	51	60	60
3	64	59	59	59	66	73
4	56	56	56	56	69	62
5	55	55	55	55	88	69
6	80	80	80	80	86	90
7	17	17	17	17	18	18
8	77	77	77	77	89	85
9	87	87	87	87	97	94
10	56	55	55	55	63	67

Tabela 5.1: Resultados para 10 navios.

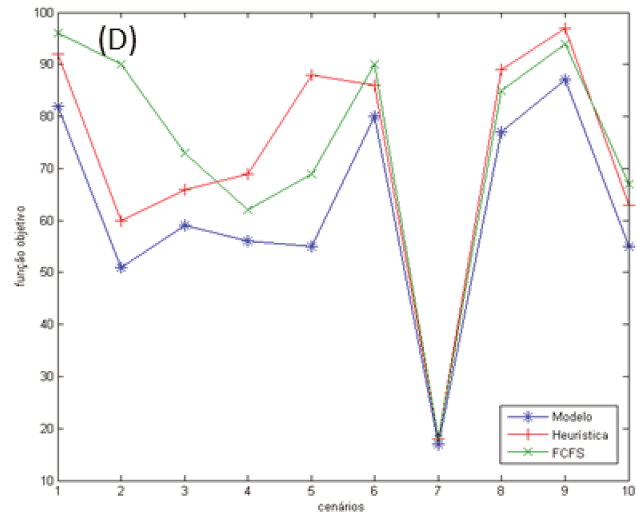
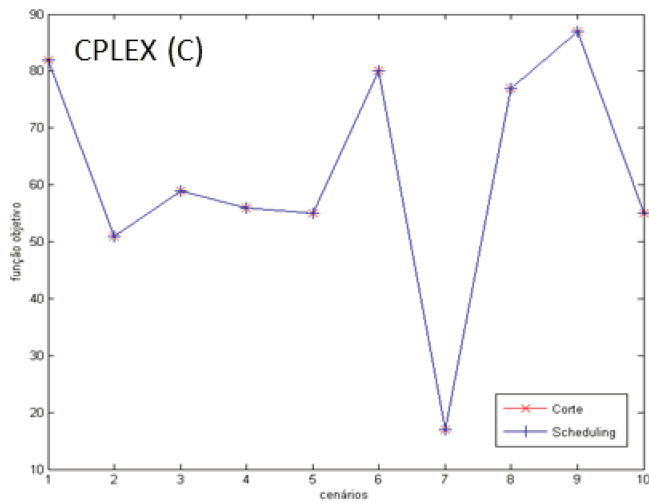
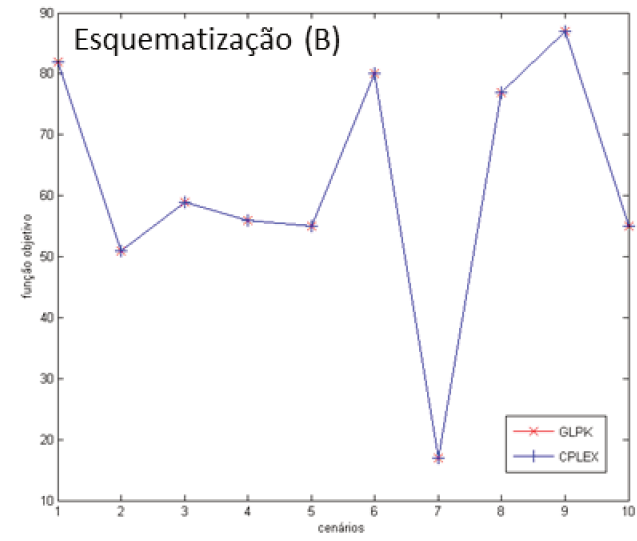
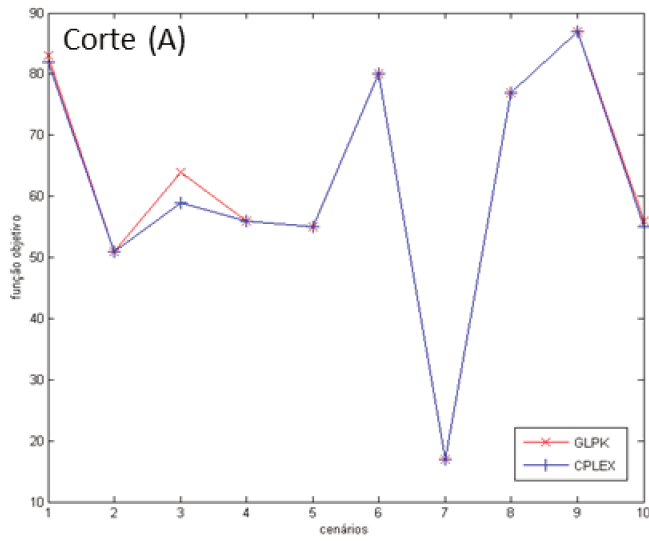


Figura 5.1: Comparações dos cenários com 10 navios.

Nos Gráficos (C) e (B) da Figura 5.1 nota-se que as curvas estão sobrepostas, pois o ótimo foi atingido em todos os casos em questão. No Gráfico (A), mesmo sem otimalidade ter sido garantida na execução do problema de corte no GLPK, é possível ver que isso não ocorreu de fato em apenas um cenário. Como o modelo de esquematização atingiu a otimalidade em todos os cenários, na Figura (D) observa-se que as heurísticas obtiveram resultados piores. Além disso, em 60% dos cenários a heurística foi melhor do que *FCFS*.

5.2 Cenários com 15 navios

Para esses cenários, o modelo de problema de corte obteve 586 restrições e 450 variáveis, das quais 435 são inteiras. Já o problema de esquematização de máquinas obteve 1111 restrições e 270 variáveis, das quais 255 são inteiras.

Na tabela 5.2 estão os resultados obtidos para os cenários com 15 navios. Em nenhum caso houve garantia de otimalidade depois de 1h.

cenário	corte GLPK	scheduling GLPK	corte CPLEX	scheduling CPLEX	<i>“Aloca-Berço”</i>	FCFS
1	277	222	220	212	244	260
2	226	211	203	203	231	260
3	171	158	153	147	174	189
4	277	274	267	263	325	306
5	203	211	189	189	219	223
6	218	217	207	207	232	238
7	248	237	226	225	280	275
8	178	170	162	160	199	201
9	244	240	232	230	268	269
10	244	240	239	238	253	257

Tabela 5.2: Resultados para 15 navios.

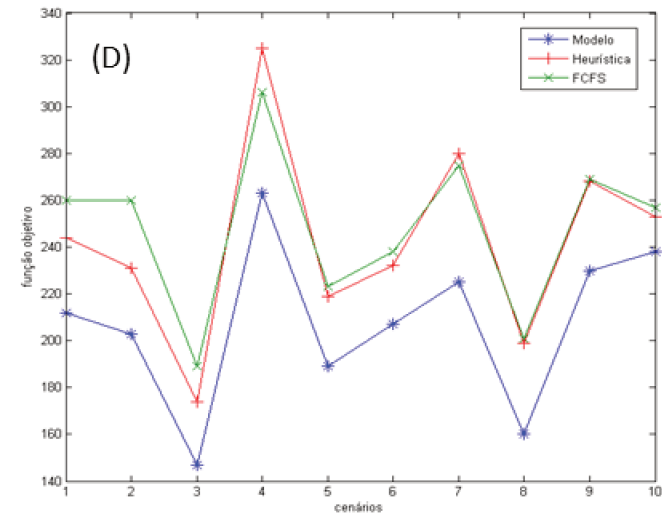
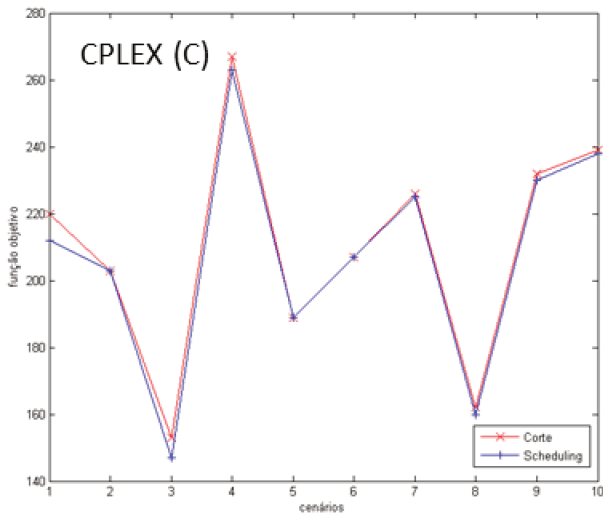
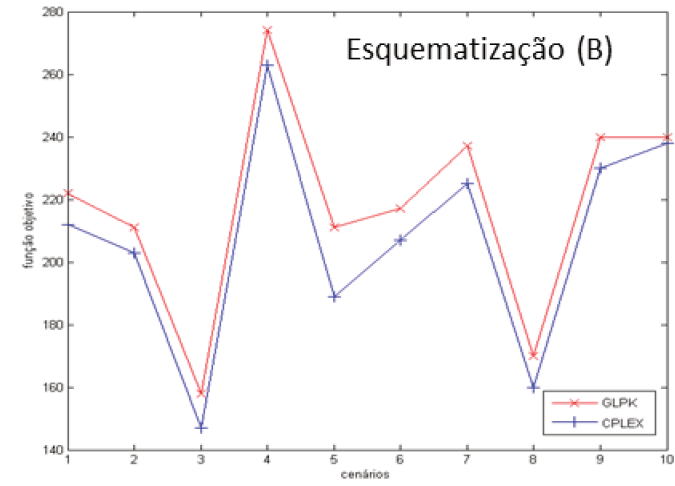
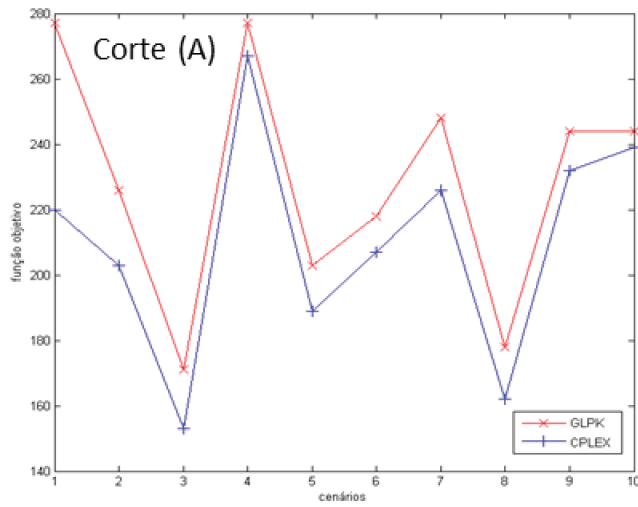


Figura 5.2: Comparações dos cenários com 15 navios.

O Gráfico (C) na Figura 5.2 mostra que, apesar de não atingir valores ótimos, os resultados obtidos para o problema de corte e para o problema de scheduling no CPLEX foram muito próximos. Nos Gráficos (A) e (B) conclui-se que para o problema de corte o CPLEX é melhor que o GLPK em todos os cenários que foram gerados. Ao comparar o melhor modelo com as heurísticas, pode-se ver na Figura (D) que o modelo de esquematização é sempre melhor do que ambas heurísticas e que *FCFS* sempre obteve resultados piores.

5.3 Cenários com 20 navios

Para esses cenários, o modelo de problema de corte obteve 1011 restrições e 800 variáveis, das quais 780 são inteiras. Já o problema de esquematização de máquinas obteve 1961 restrições e 460 variáveis, das quais 440 são inteiras.

Abaixo pode-se analisar os resultados obtidos quando os problemas são executados para diferentes cenários para 20 navios. Novamente em 1 hora não foi obtida a solução ótima e no GLPK em um cenário não foi possível encontrar uma solução factível para o problema de corte.

cenário	corte GLPK	scheduling GLPK	corte CPLEX	scheduling CPLEX	"Aloca-Berço"	FCFS
1	577	423	419	403	516	505
2	720	480	436	433	486	498
3	640	571	523	522	561	592
4	614	488	466	469	508	527
5	511	481	395	457	511	528
6	infectível	467	432	430	492	496
7	411	350	313	322	355	398
8	464	367	373	366	422	402
9	506	432	390	379	462	486
10	709	405	378	379	428	462

Tabela 5.3: Resultados para 20 navios.

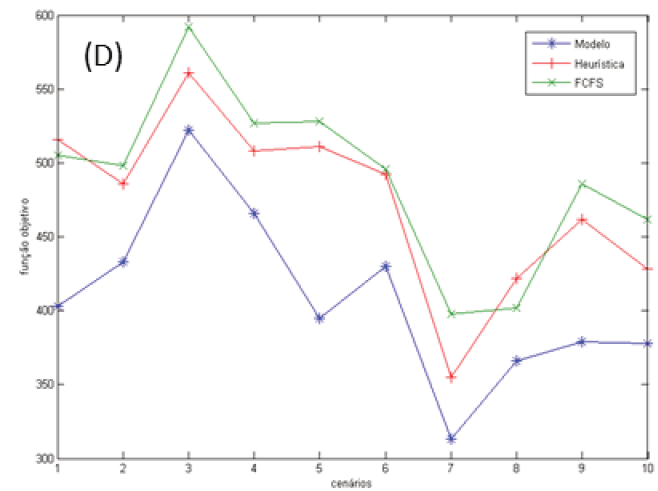
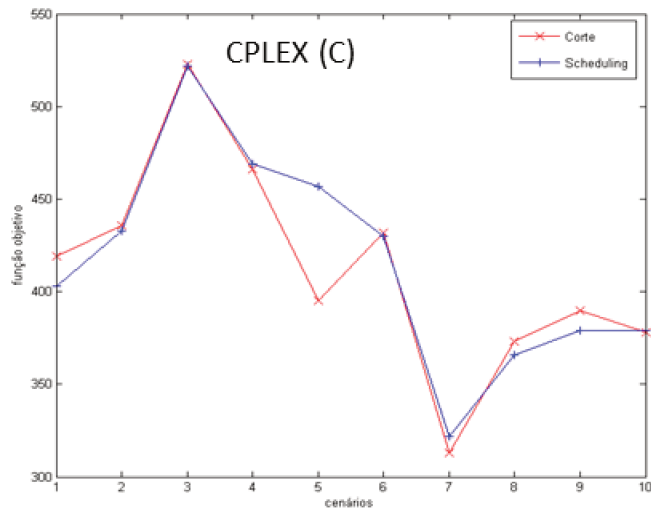
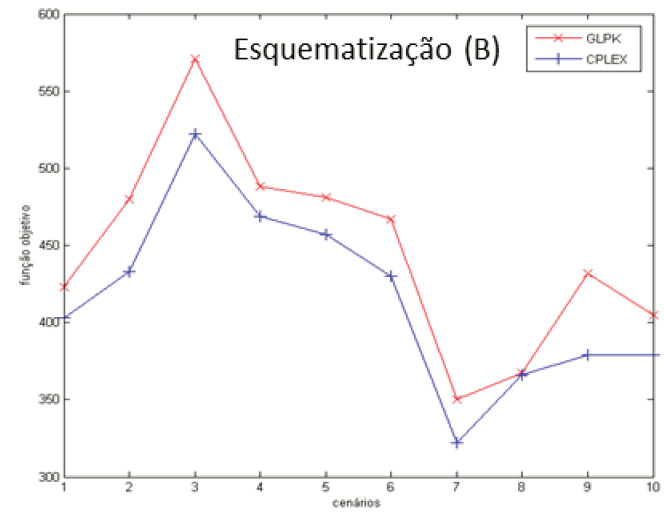
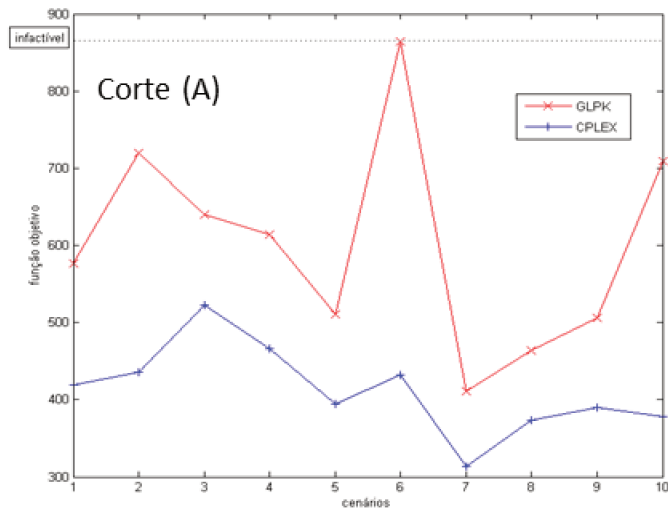


Figura 5.3: Comparações dos cenários com 20 navios.

Na comparação do Gráfico (C) da Figura 5.3 conclui-se que em 60% dos cenários o problema de esquematização obteve resultados melhores para a função-objetivo. Quando o problema de corte é resolvido, o CPLEX apresentou uma solução melhor em todos os cenários, como pode ser observado no Gráfico (A). Já para o problema de esquematização, no Gráfico (B) é fácil ver que em apenas um cenário o GLPK encontrou uma solução melhor. No Gráfico (D), observa-se que o melhor modelo sempre foi melhor do que as heurísticas e em poucos casos *FCFS* foi melhor do que a heurística.

5.4 Cenários com 25 navios

Para esses cenários, o modelo de problema de corte obteve 1576 restrições e 1250 variáveis, das quais 1225 são inteiras. Já o problema de esquematização de máquinas obteve 3076 restrições e 700 variáveis, das quais 675 são inteiras.

Na Tabela 5.4 estão os resultados para os cenários com 25 navios. Para esse tamanho, em nenhum caso o ótimo foi atingido e em 4 cenários não foi possível obter uma solução factível no GLPK para o problema de corte. Para analisar graficamente os casos infactíveis, foi atribuído um valor alto para a função-objetivo (por volta de 20% acima do maior valor obtido).

cenário	corte GLPK	scheduling GLPK	corte CPLEX	scheduling CPLEX	"Aloca-Berço"	FCFS
1	1423	945	776	772	917	961
2	1653	772	694	677	742	766
3	infactível	816	753	727	861	870
4	infactível	820	727	704	887	896
5	1392	786	764	730	830	877
6	infactível	934	739	758	866	901
7	1569	765	660	678	766	787
8	infactível	883	768	756	861	908
9	1545	879	790	790	892	927
10	1210	921	770	751	871	922

Tabela 5.4: Resultados para 25 navios.

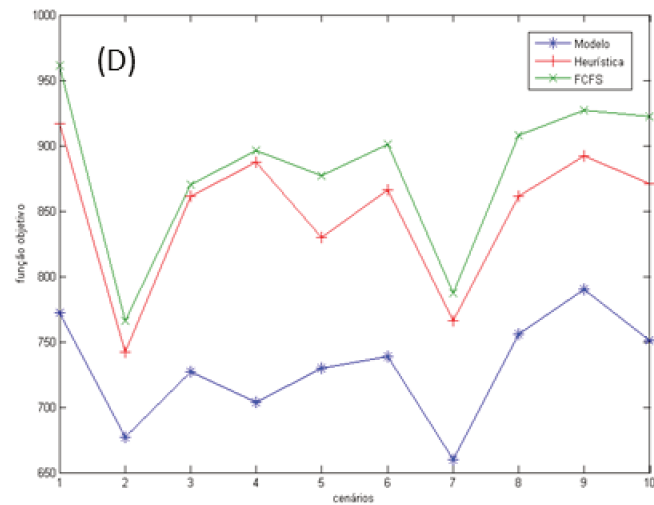
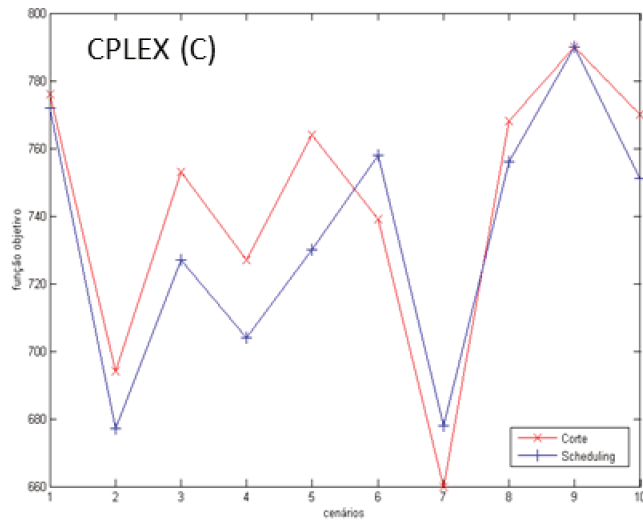
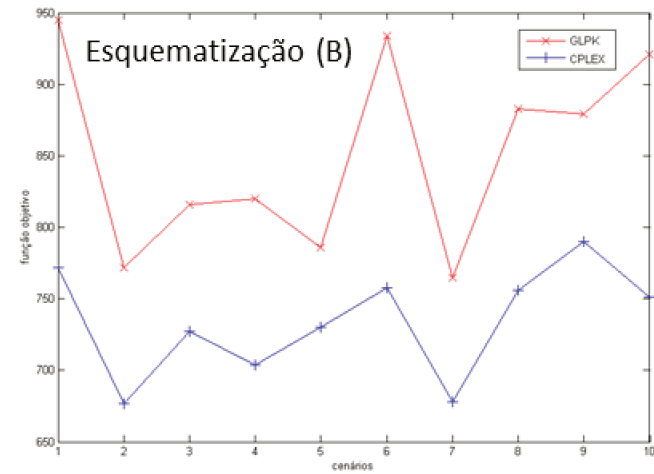
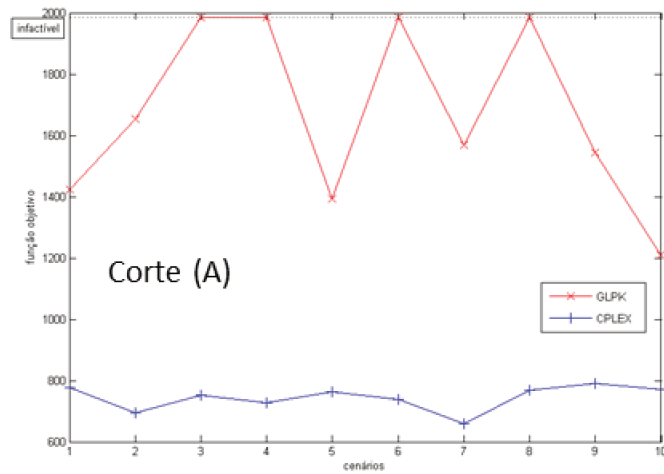


Figura 5.4: Comparações dos cenários com 25 navios.

No Gráfico (C) da Figura 5.4 nota-se que em 80% dos cenários o problema de esquematização obteve uma solução melhor ou igual ao problema de corte. Já nos Gráficos (A) e (B) verifica-se que o CPLEX obteve soluções melhores para todos os cenários em ambas modelagens. Quando o melhor modelo é comparado com as heurísticas no Gráfico (D), observa-se que a melhor solução é sempre a do modelo, seguida da heurística e por último da solução obtida pelo *FCFS*.

5.5 Cenários com 30 navios

Para esses cenários, o modelo de problema de corte obteve 2266 restrições e 1800 variáveis, das quais 1770 são inteiras. Já o problema de esquematização de máquinas obteve 4441 restrições e 990 variáveis, das quais 960 são inteiras.

Na Tabela 5.5 estão os resultados obtidos para os cenários com 30 navios.

cenário	corte GLPK	scheduling GLPK	corte CPLEX	scheduling CPLEX	"Aloca-Berço"	FCFS
1	3062	1586	1364	1312	1501	1548
2	1797	1383	1017	1008	1147	1236
3	2185	1566	1174	1222	1348	1418
4	infectível	1461	1247	1256	1355	1412
5	2796	1412	1226	1226	1456	1433
6	2036	1365	1217	1208	1307	1331
7	1866	1391	1079	1051	1144	1196
8	2058	1364	1144	1126	1129	1300
9	infectível	1349	1157	1157	1269	1302
10	infectível	1240	987	1000	1147	1203

Tabela 5.5: Resultados para 30 navios.

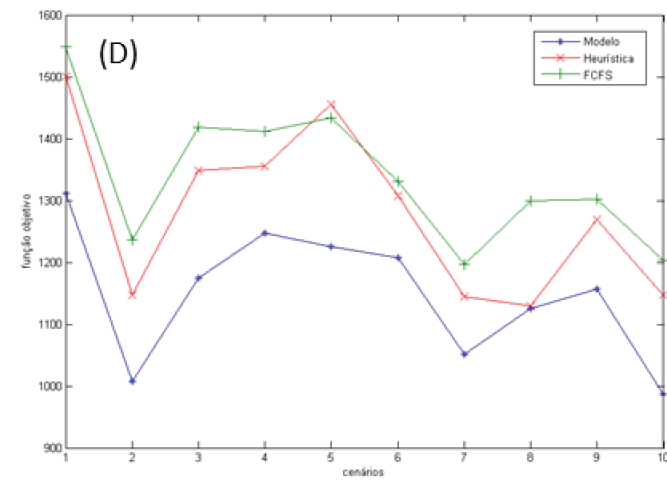
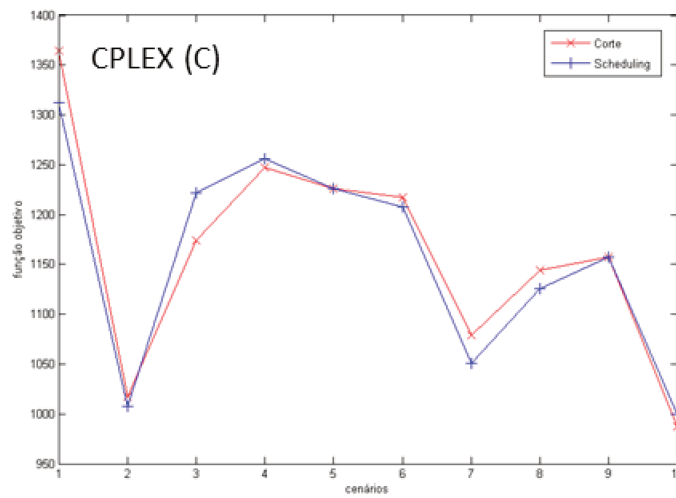
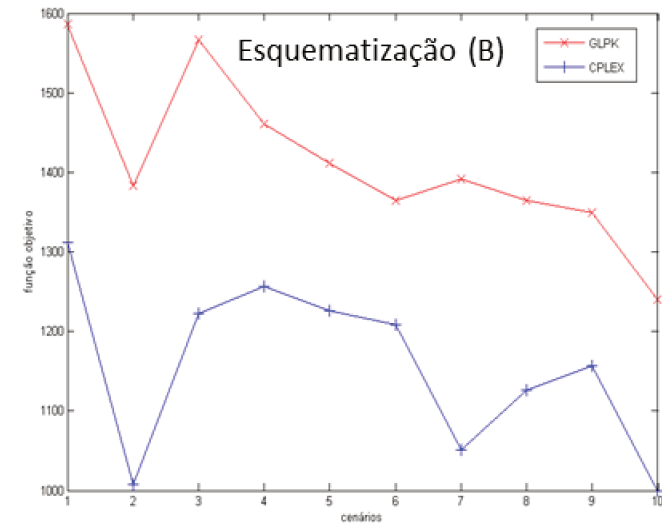
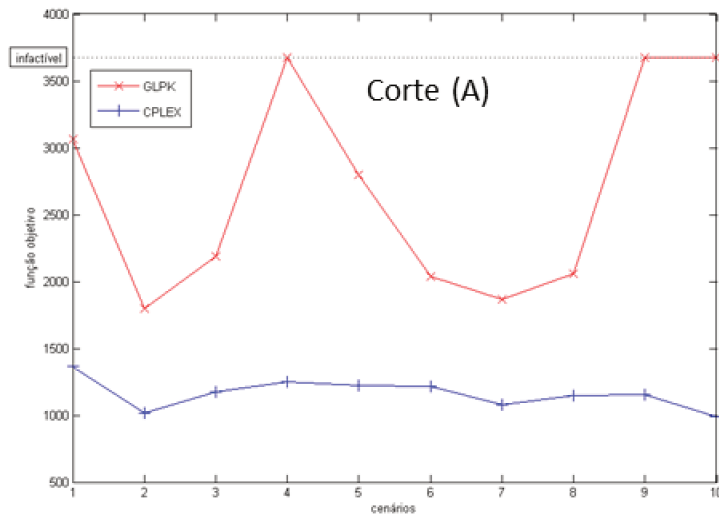


Figura 5.5: Comparações dos cenários com 30 navios.

No CPLEX, o problema de esquematização retornou valores de função objetivo melhor ou igual ao problema de corte em 60% dos cenários, como mostrado no Gráfico (C) da Figura 5.5. Nos Gráficos (A) e (B) nota-se que o desempenho do CPLEX é sempre melhor tanto para o modelo de corte quanto para o modelo de esquematização. O Gráfico (D) mostra que o melhor modelo sempre obteve um resultado melhor do que as heurísticas.

5.6 Cenários com 35 navios

Para esses cenários, o modelo de problema de corte obteve 3081 restrições e 2450 variáveis, das quais 2415 são inteiras. Já o problema de esquematização de máquinas obteve 6056 restrições e 1330 variáveis, das quais 1295 são inteiras.

Os resultados para todos os cenários com 35 navios estão na Tabela 5.6. Novamente a otimalidade não foi atingida em nenhum cenário, e para o modelo de corte no GLPK em 2 cenários não foi possível obter uma solução factível em 1 hora.

cenário	corte GLPK	scheduling GLPK	corte CPLEX	scheduling CPLEX	<i>“Aloca-Berço”</i>	FCFS
1	3118	2375	1721	1711	1925	1983
2	2644	2266	1660	1654	1878	1959
3	4505	2074	1688	1711	1926	1962
4	3734	2041	1568	1570	1823	1922
5	infectível	2114	1683	1633	1798	1831
6	3478	2113	1659	1671	1903	1999
7	3223	2295	1559	1543	1806	1918
8	3415	2375	1777	1801	2004	2014
9	3515	2369	1794	1756	1888	1898
10	infectível	2206	1710	1670	1941	1987

Tabela 5.6: Resultados para 35 navios.

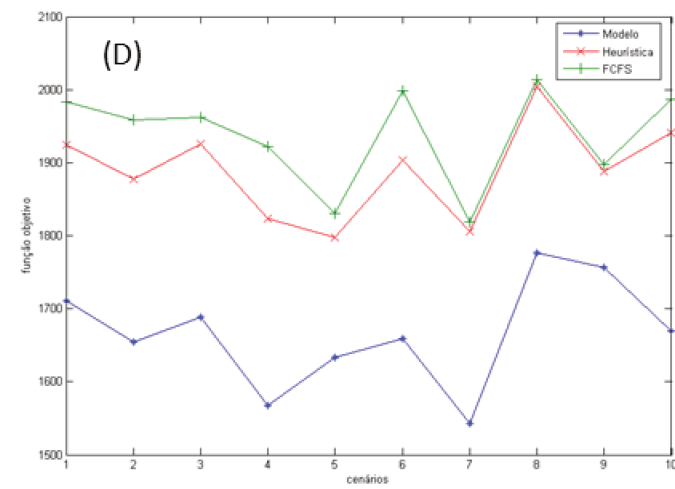
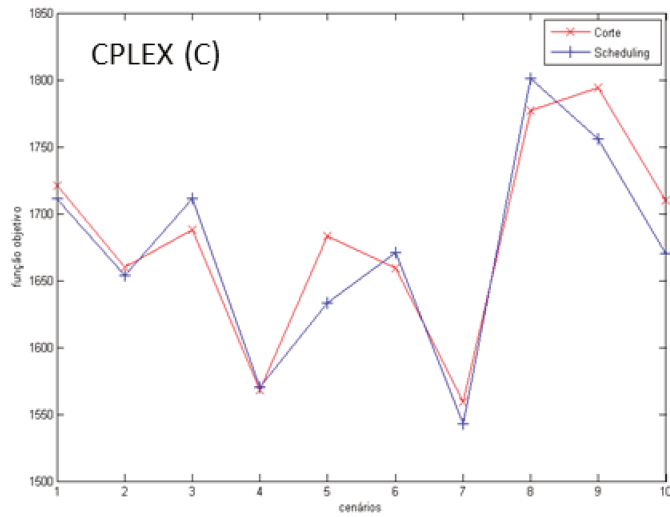
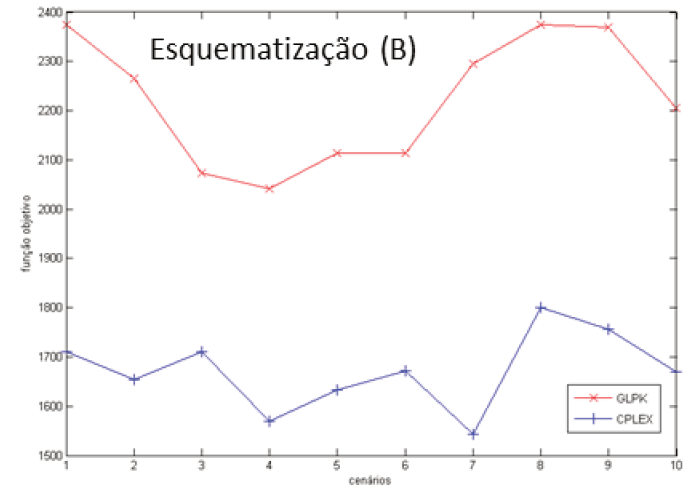
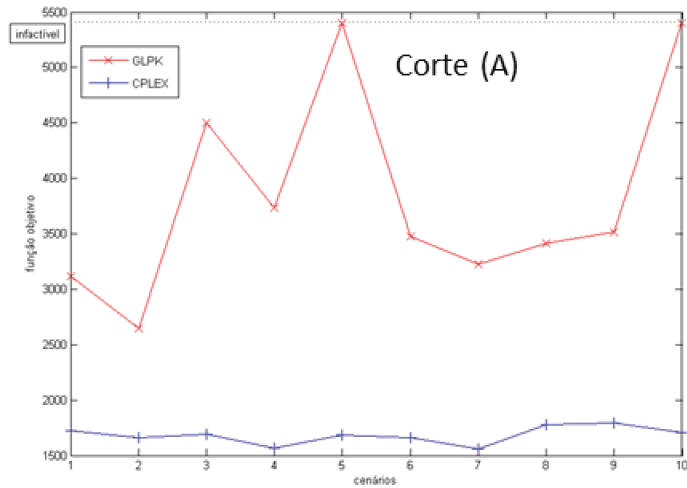


Figura 5.6: Comparações dos cenários com 35 navios.

Observando o Gráfico (C) da Figura 5.6, conclui-se que em 60% dos cenários no CPLEX o modelo de scheduling obteve uma solução com valor da função-objetivo melhor. Nos Gráficos (A) e (B) tem-se que novamente que o CPLEX obteve soluções muito melhores para todos os cenários os modelos de corte e esquematização. Pelo Gráfico (D), o pior resultado foi de *FCFS*, seguido da heurística e, então, do melhor modelo.

5.7 Cenários com 40 navios

Para esses cenários, o modelo de problema de corte obteve 4021 restrições e 3200 variáveis, das quais 3160 são inteiras. Já o problema de esquematização de máquinas obteve 7921 restrições e 1720 variáveis, das quais 1680 são inteiras.

Para os cenários com 40 navios, os resultados estão na Tabela 5.7. Mais uma vez o ótimo não foi atingido em nenhum caso e em 3 cenários o modelo de corte não encontrou solução factível em 1 hora.

cenário	corte GLPK	scheduling GLPK	corte CPLEX	scheduling CPLEX	<i>“Aloca-Berço”</i>	FCFS
1	infectível	3330	2085	2116	2262	2322
2	infectível	3413	2369	2397	2655	2770
3	5023	3400	2264	2167	2382	2460
4	infectível	3056	2132	2132	2328	2420
5	4376	3516	2297	2259	2497	2578
6	5013	3144	2265	2217	2557	2673
7	5469	3151	2276	2221	2583	2611
8	4778	2829	2161	2182	2487	2589
9	4907	2962	2298	2309	2547	2633
10	5497	3068	2229	2164	2506	2595

Tabela 5.7: Resultados para 40 navios.

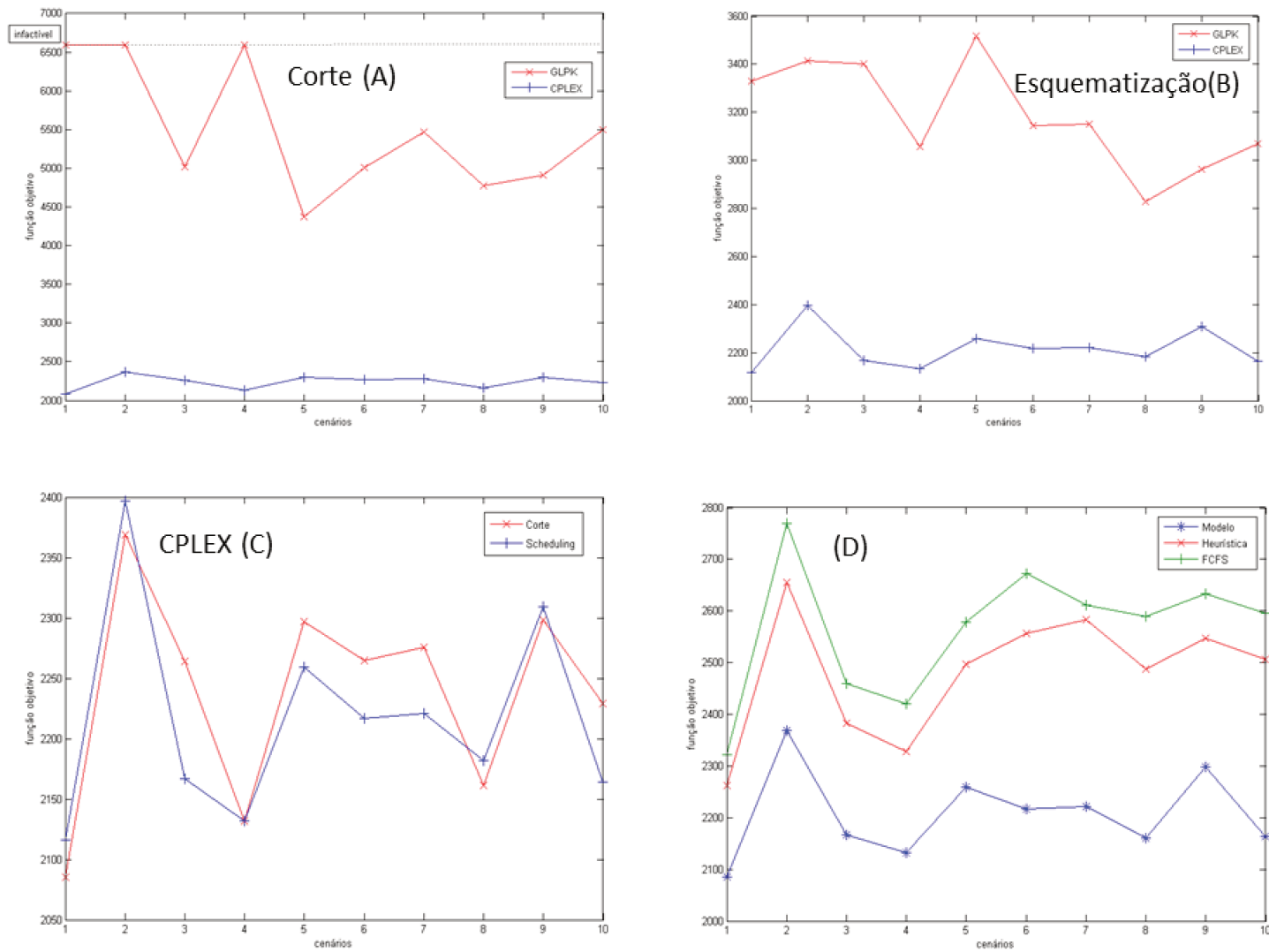


Figura 5.7: Comparações dos cenários com 40 navios.

Comparando o desempenho do problema de corte e do modelo de esquematização no CPLEX nota-se pelo Gráfico (C) da Figura 5.7 que em 60% dos cenários o segundo obteve para a função-objetivo um valor melhor ou igual do que o primeiro. Tanto para o modelo de corte quanto para o modelo de esquematização vemos nos Gráficos (A) e (B) que o CPLEX encontrou soluções melhores. Comparando o melhor modelo com as heurísticas, novamente no Gráfico (D) é mostrado que esse foi melhor que a heurística.

Capítulo 6

Conclusão

O objeto de estudo do presente trabalho foi o Problema de Alocação de Berços: qual a melhor maneira de alocar os navios que chegam ao terminal de modo a minimizar o tempo de espera desses. Após um abrangente estudo sobre as variações que estão disponíveis na literatura, dois modelos matemáticos foram adaptados e uma heurística foi construída. O primeiro modelo foi adaptado a partir de um problema de corte tri-dimensional. O segundo é baseado em um problema de esquematização de máquinas. A heurística aloca os navios considerando qual berço está com um horizonte de planejamento menor. Além disso, para fins de comparação, também foi implementado o método *First Come, First Serve (FCFS)*, que é utilizado na prática no terminal.

Os parâmetros de horário de chegada e tempo de processamento dos navios foram gerados e usados para resolver o problema por esses três métodos. Os dois modelos foram executados usando tanto o GLPK quanto o CPLEX, e foi usado como limitante o tempo, estipulado em 1 hora de execução.

A otimalidade só foi garantida para os cenários com 10 navios. Pelos gráficos e tabelas apresentados no Capítulo 5, o desempenho do CPLEX foi notavelmente melhor para ambos modelos, apresentado o melhor resultado de função-objetivo (mesmo sem garantia de otimalidade). Em alguns cenários, o GLPK não conseguiu sequer obter uma solução factível dentro do tempo estipulado. Quando os modelos foram comparados, a diferença apresentada foi pequena, mas em pouco mais da metade dos casos o modelo de esquematização retornou um valor melhor para a função-objetivo minimizada.

Depois do resultado do modelo, está o resultado obtido pela heurística.

Também é possível ver que o *FCFS* em quase todas as vezes foi o método que teve o pior resultado. De modo geral, pode-se concluir que uma maneira mais eficaz de fazer o planejamento das operações de atracação no terminal foi obtida.

Aqui apenas foi procurada uma maneira mais eficiente de organizar o horário e a posição de atracação dos navios ao longo do cais. No caso do Terminal de Praia Mole, as operações portuárias ds Vale vão além disso. Um planejamento mais abrangente envolve também buscar otimizar o funcionamento das esteiras que transportam o carvão até os pátios de estocagem e a formação dos trens que levarão o carvão até seu destino final.

Referências Bibliográficas

- [1] Andrade, H. R. B. *Análise do Pátio Ferroviário de Tubarão no atendimento ao Carvão*. Dissertação de mestrado., 2008.
- [2] Bierwirth, C., Meisel, F. *A survey of berth allocation and quay crane scheduling problem in container terminals*. European Journal of Operations Research 202(3): 615-627, 2010.
- [3] Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J., Lusby, R. *Models for the discrete berth allocation problem: a computational comparison*. Transportation Research Part E, 2009.
- [4] Chen, L., Bostel, N., Dejax, P., Cai, J., Xi, L. *A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal*. European Journal of Operation Research 181 (1). 40-58., 2007.
- [5] Chen, C. Y., Chao, S. L. *A time-space network model for allocating yard storage space for export containers* Transportation Planning Journal Vol. 33 N_o 2. 227-248, 2004.
- [6] Chen, C. S., Lee, S. M., Shen, Q. S. *An analytical model for the container loading problem*. European Journal og Operational Research 80 (1995) 69-76.
- [7] Choi, B.-C., Lee, K., Leung, J Y. -T, Pinedo, M. L., Briskorn, D. *Container scheduling: complexity and algorithms*. Production and Operations Management Vol. 21, N_o 1. pp. 115-128., 2012.
- [8] Cruz, M. M. C., Farias, F. N., Souza, R. G., Vaz, R. G., Silva, S. *Estudo por simulação da operação de granéis sólidos do Porto de Tubarão*.

- [9] Dai, J., Lin, W., Moorthy, R., Teo, C.-P. *Berth allocation planning optimization in container terminal*. 2004.
- [10] Fioroni, M. M., Pompermayer, F. M., Franzese, L. A. G. *Estudo estratégico de dimensionamento da infra-estrutura do pátio ferroviário de Tubarão através de simulação*. XVIII Congresso de Pesquisa e Ensino em Transportes.
- [11] Giallombardo, G., Moccia, L., Salani, M., Vacca, I. *The tactical berth allocation problem with quay crane assignment and transshipment-related quadratic yard costs*. Proceedings of the European Transport Conference (ETC), pp. 1-27., 2008.
- [12] Guam, Y., Cheung, R. K. *The berth allocation problem: models and solution methods*. OR Spectrum 26 (1) 75-92, 2004.
- [13] Imai, A., Chen, H. C., Nishimura, E., Papadimitriou, S. *The simultaneous berth and quay allocation problem*. Transportation Research Part E 44(5). 990-920., 2008a.
- [14] Imai, A., Nishimura, E., Papadimitriou S. *The dynamic berth allocation problem for a container port*. Transportation Research Part B 35 401-417, 2001.
- [15] Imai, A., Sun, X., Nishimura, E., Papadimitriou, S. *Berth allocation in a container port: using a continuous location space approach*. Transportation Research Part B 39 (3), 199-221, 2005.
- [16] Imai, A., Zhang, J. T., Nishimura, E., Papadimitriou, S. *The berth allocation problem with service time and delay time objectives*. Maritime Economics and Logistics 9 (4), 269-290, 2007.
- [17] Kim, K. H., Moon, K. C. *Berth Scheduling by simulated annealing*. Transportation Research Part B 37(6). 541-560, 2003.
- [18] Liang, C., Huang, Y., Yang, Y. *A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning*. Computers and Industrial Engineering 56 (3). 537-547., 2008.
- [19] Lim, A. *The berth planning problem*. Operations Research Letters 22(2-3): 105-110., 1998.

- [20] Liu, J., Wan, Y.-W., Wang, L. *Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessels departures*. Naval Research Logistics 53 (1). 60-74., 2006.
- [21] Lopes, A. T., Schulz, V. M. L., Mauri, G. R. *Grasp com path relinking para o problema de alocação de berços*.
- [22] Mauri, G. R., Oliveira, A. C. M., Lorena, L. A. N. *Resolução do problema de alocação de berços através de uma técnica de geração de colunas*.
- [23] Oliveira, R. M., Mauri, G. R., Lorena, L. A. N. *Clustering Search aplicado ao problema de alocação de berços*.
- [24] Patricio, M., Botter, R. C. *Análise de regras de atracação de navios em terminais de containers*.
- [25] Silva, V. M. D., Coelho, A. S., Mayerle, S. F., Abreu, L. F. *Heurística para a resolução do problema de alocação de navios em berços usando algoritmos genéticos*.
- [26] Umang, N., Bierlare, M., Vacca, I. *Exact and heuristic methods to solve the berth allocation problem in bulk ports*. Report Transport and Mobility Laboratory, 2012.
- [27] Yang, C., Wang, N. *A solution method for berth-quay crane allocation problem*. Transportation and Logistics College, 2010.
- [28] Zhang, Y. *Dynamic quay crane allocation*. Dissertations and Theses Collection. Paper 62., 2010.
- [29] Vale *Informações do Porto de Tubarão*.

Apêndice A

Dados gerados

e	p	e	p	e	p	e	p	e	p
9	10	23	7	7	7	20	8	11	4
3	7	22	10	15	6	14	3	21	9
22	7	17	10	10	12	24	9	15	7
12	10	7	7	11	9	10	6	1	7
16	7	6	9	13	10	3	8	25	6
3	8	12	7	3	6	6	13	9	11
9	8	2	10	16	9	14	11	20	8
10	12	20	8	2	5	19	10	11	11
6	10	21	8	1	7	5	7	20	7
21	9	25	6	11	8	18	7	18	8

Tabela A.1: Dados para 10 navios (I)

e	p	e	p	e	p	e	p	e	p
18	9	12	8	5	10	19	7	21	9
7	9	2	7	20	9	1	7	16	8
21	6	5	6	10	7	6	6	5	10
7	10	19	5	4	8	7	8	20	8
13	9	1	6	23	6	17	10	22	6
21	10	10	4	18	10	7	8	7	8
16	9	21	5	8	9	11	11	19	2
23	12	23	10	2	9	5	8	6	10
24	6	13	7	7	9	7	10	12	8
9	8	18	6	24	10	14	9	3	6

Tabela A.2: Dados para 10 navios (II)

e	p	e	p	e	p	e	p	e	p
6	6	23	8	22	12	24	11	18	7
5	6	1	11	24	8	3	11	11	12
8	5	9	5	24	5	5	9	0	7
10	8	8	12	11	4	8	7	18	6
3	11	16	7	16	5	25	6	2	6
3	8	6	7	17	9	2	12	1	10
17	9	21	9	7	8	6	12	21	10
20	7	16	3	21	7	21	7	4	6
17	11	5	7	10	10	9	7	3	7
6	9	15	13	2	9	21	11	25	9
16	11	10	7	8	8	25	10	4	8
22	2	23	10	14	7	7	6	25	7
10	10	10	9	7	6	15	10	7	6
5	6	6	8	13	7	4	8	12	9
22	7	20	6	1	7	13	10	14	11

Tabela A.3: Dados para 15 navios (I)

e	p	e	p	e	p	e	p	e	p
3	8	3	9	15	10	15	6	3	8
6	10	4	6	17	7	12	6	15	8
13	5	15	8	1	10	3	10	3	8
15	7	18	6	6	6	9	5	21	10
8	10	14	5	19	8	8	3	1	9
16	10	10	10	3	7	22	11	6	8
5	8	4	8	19	7	9	7	8	7
1	8	8	10	24	10	22	7	20	8
3	7	11	11	24	3	6	9	14	7
1	4	17	9	10	6	6	10	4	10
3	11	23	8	10	6	7	8	17	12
23	9	13	7	16	9	13	7	13	8
21	9	10	10	2	10	14	7	12	9
21	9	10	8	23	7	5	9	22	8
20	8	2	6	20	8	17	9	1	7

Tabela A.4: Dados para 15 navios (II)

e	p	e	p	e	p	e	p	e	p
16	8	4	8	8	9	17	12	14	9
4	7	21	7	12	7	22	9	2	7
6	10	7	9	10	9	5	9	4	8
12	7	6	6	6	9	22	7	18	8
18	7	20	6	13	10	5	5	14	9
10	13	2	7	5	6	12	8	3	8
17	9	22	9	16	9	15	10	10	6
20	5	13	8	6	10	6	10	21	8
2	10	15	9	10	11	8	7	2	9
1	10	6	6	19	9	9	8	10	7
19	7	18	7	15	8	19	8	1	8
9	6	8	10	18	9	3	8	7	5
5	6	14	6	6	6	3	7	12	6
21	7	10	10	22	7	2	7	3	6
15	5	9	7	10	7	5	7	14	6
23	7	1	7	6	8	20	9	23	6
0	7	20	10	17	12	4	8	12	11
5	9	16	7	17	8	9	2	20	7
4	7	14	8	10	8	1	8	2	10
25	6	2	9	9	3	0	10	3	8

Tabela A.5: Dados para 20 navios (I)

e	p	e	p	e	p	e	p	e	p
11	7	24	8	13	9	18	5	6	9
18	5	24	11	16	8	23	4	12	9
7	9	1	7	12	7	1	6	1	9
11	9	7	11	1	8	3	4	6	9
11	6	10	5	24	11	12	8	8	10
7	4	1	8	23	7	8	12	2	6
10	8	16	5	3	8	6	8	15	9
19	9	22	7	14	6	11	10	23	8
7	6	16	7	2	6	23	9	9	9
7	4	21	6	0	9	17	10	24	4
9	7	24	4	3	3	2	7	18	7
10	6	20	11	22	6	11	11	13	6
6	10	2	7	23	10	5	10	24	8
23	6	16	12	14	7	25	8	24	6
20	10	21	5	24	8	19	9	25	9
12	7	6	7	9	7	15	9	20	6
15	8	12	9	3	7	7	7	9	7
15	10	21	9	17	5	25	5	17	7
17	7	25	9	5	8	11	8	11	6
5	6	23	6	14	6	8	6	8	7

Tabela A.6: Dados para 20 navios (II)

e	p	e	p	e	p	e	p	e	p
9	7	16	7	13	7	19	10	19	6
3	8	20	10	11	9	17	6	2	5
7	10	6	8	12	7	24	11	11	8
11	13	25	12	10	8	7	7	19	7
19	2	21	7	8	9	17	9	17	9
8	6	1	4	17	11	0	6	9	9
19	10	16	8	8	9	24	9	16	6
6	9	6	5	18	4	6	7	7	4
20	6	22	10	14	7	4	12	9	7
8	5	14	7	5	7	6	9	6	7
8	7	16	8	5	6	13	6	20	8
12	9	20	7	7	9	7	5	10	9
17	7	9	11	20	8	11	6	14	8
7	12	11	8	11	8	17	8	8	8
25	7	8	7	18	6	24	4	3	8
4	10	1	8	15	11	17	7	7	3
24	8	13	6	2	9	23	9	12	7
17	7	7	8	14	9	23	11	6	10
2	9	13	10	18	4	5	10	3	11
12	9	5	6	10	8	2	10	7	10
4	7	25	6	25	6	11	10	6	8
2	10	20	7	22	10	2	10	6	6
1	8	2	8	19	8	16	4	14	5
20	8	17	10	1	9	4	10	12	9
21	8	18	10	20	4	24	6	16	9

Tabela A.7: Dados para 25 navios (I)

e	p	e	p	e	p	e	p	e	p
24	10	24	8	5	6	3	8	20	9
5	6	4	4	1	8	12	12	7	10
0	7	9	4	8	6	4	10	1	7
2	5	4	5	21	10	20	8	18	9
8	9	17	9	19	10	10	7	14	7
22	11	18	9	14	8	3	5	11	11
24	7	6	9	15	12	2	10	3	9
18	5	12	7	8	10	11	7	23	5
10	5	5	3	21	6	5	8	2	9
13	7	19	13	13	10	19	8	5	3
15	6	24	6	18	6	12	6	24	4
7	10	4	4	23	4	15	12	8	7
15	12	23	8	12	9	3	9	2	10
15	7	21	5	12	7	4	7	17	10
20	7	13	9	21	10	18	8	20	12
3	10	5	8	9	10	24	5	18	10
11	6	2	12	24	8	5	6	4	10
4	11	8	9	13	5	19	7	23	10
25	8	16	6	3	8	11	7	3	3
3	11	13	8	1	9	2	8	0	8
6	10	21	6	1	7	2	8	2	9
19	6	4	5	12	9	4	7	4	8
6	8	8	9	12	11	15	7	4	5
6	9	18	9	22	8	1	7	19	7
12	8	5	10	3	9	14	6	3	11

Tabela A.8: Dados para 25 navios (II)

e	p	e	p	e	p	e	p	e	p
13	9	23	7	17	8	1	8	10	7
8	11	15	7	2	9	18	10	10	9
11	7	17	5	11	6	14	6	18	9
12	6	21	7	4	11	15	10	4	11
6	7	17	10	10	9	12	11	19	13
17	9	15	11	18	10	20	10	12	10
7	10	21	5	1	9	17	8	11	9
12	10	24	5	23	6	22	9	0	9
25	7	0	8	13	10	7	7	8	10
1	11	24	10	11	2	4	7	15	10
20	6	24	9	8	8	1	7	2	10
9	11	19	9	14	8	4	11	2	6
6	11	20	7	5	9	18	8	14	6
5	7	3	7	24	7	2	8	21	7
8	9	8	7	1	10	7	12	2	10
15	8	23	8	21	7	4	7	24	10
22	9	8	8	24	10	8	7	4	11
23	7	15	11	6	12	10	5	0	9
7	8	17	4	6	6	19	11	9	8
17	8	25	8	7	8	3	10	2	8
13	7	13	7	18	6	7	9	22	9
12	8	14	6	19	4	21	11	18	7
12	6	15	7	6	10	7	11	12	8
7	10	3	7	17	8	11	8	24	8
5	10	14	11	10	6	11	4	12	6
24	8	6	9	17	10	12	4	5	8
24	9	19	6	8	9	19	9	22	9
19	9	18	6	23	9	6	8	15	7
10	8	12	10	19	7	21	8	18	4
12	6	13	10	24	11	14	8	3	4

Tabela A.9: Dados para 30 navios (I)

e	p	e	p	e	p	e	p	e	p
13	8	22	8	25	4	1	6	2	7
11	7	5	7	13	14	20	9	14	9
24	9	24	7	19	7	4	4	10	8
23	7	23	11	12	8	24	8	8	6
3	10	2	8	24	9	6	10	13	7
17	10	8	8	4	9	0	5	14	6
17	11	6	3	9	7	1	9	4	6
8	9	20	6	18	7	13	8	4	11
8	5	18	11	8	8	8	10	13	9
4	11	24	9	8	10	2	8	6	7
11	7	13	6	18	9	8	6	16	8
16	7	14	10	8	5	6	8	9	8
12	10	8	6	3	7	9	10	18	4
21	8	2	7	19	10	24	8	22	6
21	8	22	6	17	9	3	9	17	4
23	10	23	11	6	8	2	9	3	8
21	9	0	6	9	10	9	8	6	3
2	4	10	8	17	7	22	8	24	6
9	9	4	9	10	8	24	7	4	8
12	5	8	7	23	10	4	8	1	9
19	8	17	8	15	7	14	10	10	10
13	10	18	13	20	13	21	7	12	7
19	8	0	5	22	7	13	6	13	11
19	9	2	10	10	7	19	10	19	5
11	9	8	7	0	7	15	9	11	8
0	7	6	7	15	6	18	8	1	5
14	9	21	11	17	9	8	8	2	5
21	10	3	8	8	6	2	6	25	9
15	7	23	7	15	7	19	10	16	7
10	11	23	8	16	7	21	6	8	5

Tabela A.10: Dados para 30 navios (II)

e	p	e	p	e	p	e	p	e	p
16	10	20	8	19	7	13	8	7	5
10	13	9	6	9	9	19	5	11	8
13	6	22	8	13	11	15	11	16	9
22	9	5	8	8	11	6	10	10	10
18	9	7	5	14	9	21	8	16	5
15	7	14	10	17	10	20	5	9	9
21	9	6	8	17	11	5	10	4	9
2	6	15	8	14	10	13	4	24	10
2	8	18	7	9	7	1	9	19	8
19	7	1	8	15	6	2	7	24	9
3	9	2	9	6	13	3	9	2	7
20	6	11	6	25	9	9	9	19	10
12	3	15	11	17	10	14	11	9	8
14	10	19	7	12	9	21	7	10	8
1	8	25	4	22	9	8	12	18	11
20	7	8	8	9	7	12	7	5	8
6	8	9	8	6	5	4	8	17	11
10	8	11	9	10	8	19	5	6	8
15	11	18	9	7	5	20	6	9	8
16	10	14	7	8	6	10	6	3	5
24	10	3	10	1	10	6	8	23	12
10	10	1	6	8	7	9	6	2	6
20	7	7	11	23	9	12	8	2	5
17	8	8	6	16	10	20	9	10	6
19	5	15	7	20	4	19	8	1	11
3	8	21	8	20	5	13	10	17	9
1	8	14	11	15	8	9	5	18	8
9	10	16	9	18	10	17	8	22	6
22	8	2	10	18	10	10	7	7	6
8	7	7	12	13	8	1	6	10	7
9	9	24	9	1	8	7	10	22	6
4	9	12	6	19	9	18	9	12	8
21	6	10	6	6	6	8	7	9	8
22	8	22	4	19	10	7	6	13	9
10	11	8	8	14	7	13	6	20	6

Tabela A.11: Dados para 35 navios (I)

e	p	e	p	e	p	e	p	e	p
4	7	11	7	21	8	11	11	1	8
16	9	15	10	23	10	14	9	25	7
8	9	22	6	24	9	6	7	11	7
8	9	21	8	11	9	23	8	13	9
1	7	21	6	18	8	8	10	6	7
15	5	17	10	2	8	14	9	1	7
12	12	13	9	18	8	17	12	5	10
13	8	3	7	4	8	17	6	21	9
5	7	2	11	1	9	17	5	5	10
7	8	21	5	11	9	8	8	1	11
23	7	12	8	4	8	11	8	18	6
1	8	23	9	7	10	20	10	21	8
21	9	4	5	5	9	6	9	18	7
19	7	4	6	15	10	13	7	14	8
13	3	8	9	7	7	18	8	20	10
23	8	4	6	18	10	9	10	22	7
13	7	16	7	15	7	16	8	21	6
5	10	20	11	8	12	21	7	22	10
6	8	6	12	12	10	4	9	20	5
21	8	19	9	20	8	0	7	20	6
21	8	4	6	15	11	3	4	2	9
11	7	12	10	23	4	24	10	9	7
2	7	19	9	9	6	12	9	23	9
15	8	17	5	5	4	2	8	13	8
6	9	4	8	20	10	15	6	7	9
6	15	0	6	7	9	20	8	22	9
8	9	0	8	17	9	5	8	11	7
18	8	13	10	23	9	0	9	24	6
23	8	12	11	4	8	3	8	8	10
20	11	21	8	8	8	18	10	8	11
24	7	7	7	14	5	7	6	5	10
20	4	2	12	6	9	16	8	9	8
1	7	16	5	22	4	11	6	1	6
3	11	24	4	1	6	25	12	2	7
5	8	16	9	3	11	10	10	11	8

Tabela A.12: Dados para 35 navios (II)

e	p	e	p	e	p	e	p	e	p
3	7	2	10	14	8	4	7	18	10
14	9	15	12	21	11	24	9	19	9
4	7	22	9	25	8	22	8	2	5
6	4	24	6	21	7	3	5	1	10
7	8	9	9	17	7	7	8	5	9
11	6	19	6	11	7	15	11	5	7
10	8	13	9	17	9	24	9	18	10
16	7	22	3	7	6	22	9	23	8
4	9	1	10	9	9	15	9	8	9
19	8	10	9	9	9	8	8	14	12
8	7	3	10	20	6	11	6	5	8
12	6	13	4	4	10	24	8	9	8
6	5	6	8	19	7	13	12	3	4
7	6	6	9	21	6	25	4	22	8
2	3	12	7	13	10	14	8	4	5
17	10	6	6	4	9	15	12	22	11
7	12	11	11	20	2	20	8	7	5
25	8	9	9	15	9	3	5	16	9
1	6	19	6	3	6	22	10	4	8
16	12	21	11	16	9	22	8	16	9
10	7	9	7	0	7	6	9	23	11
13	9	1	9	12	9	8	6	20	10
23	10	7	10	4	9	5	6	10	10
22	7	20	9	2	7	13	10	10	9
12	9	19	9	1	5	20	8	5	6
13	10	8	7	4	7	20	9	12	7
11	8	22	9	5	7	8	7	8	4
11	10	1	8	10	8	23	4	10	8
10	9	19	9	9	7	16	7	23	7
16	8	12	9	22	11	7	8	8	9
22	9	21	9	6	9	14	8	2	11
8	8	4	7	13	10	15	9	16	5
13	5	21	6	23	10	22	8	10	7
2	10	4	10	14	8	11	6	21	9
19	9	13	7	20	8	18	10	18	7
18	10	16	8	23	9	1	6	4	12
20	10	5	8	19	7	24	8	24	6
18	6	11	8	1372	9	12	8	3	9
1	2	21	7	19	6	16	9	18	8
5	8	22	9	22	10	23	6	5	6

Tabela A.13: Dados para 40 navios (I)

e	p	e	p	e	p	e	p	e	p
2	6	1	8	14	6	16	5	9	9
15	7	11	9	25	6	14	9	21	4
7	9	7	10	16	4	6	7	10	9
22	5	15	10	21	7	13	6	4	9
8	5	22	9	2	14	14	9	9	10
5	10	16	9	18	10	18	5	18	7
24	6	15	9	2	8	3	10	3	9
4	8	13	6	15	9	10	9	3	7
11	8	22	7	7	10	6	7	13	7
21	9	3	10	16	9	20	6	19	10
21	8	0	11	0	6	23	8	2	7
8	5	17	9	6	6	12	8	8	6
14	6	12	8	12	8	24	11	18	5
6	9	16	4	17	7	4	5	22	10
20	9	1	8	15	8	13	9	4	11
9	6	1	7	13	10	7	9	1	5
18	9	22	5	2	9	8	9	5	6
22	7	5	9	0	7	12	9	6	10
4	8	6	7	5	9	17	8	10	7
14	7	25	10	18	6	13	8	20	8
16	6	24	11	4	8	14	4	18	10
15	8	17	3	7	12	24	8	0	10
23	8	17	6	22	8	14	8	16	6
13	7	8	9	10	9	12	8	11	8
20	9	24	7	2	8	14	8	23	5
6	9	3	4	25	11	11	8	13	11
23	5	22	11	24	7	19	6	22	7
4	11	0	11	20	10	14	9	22	9
25	11	18	8	7	9	12	12	15	8
3	11	19	8	10	8	2	9	18	8
7	12	21	8	11	5	3	4	6	6
7	7	24	8	24	11	3	9	12	8
14	8	21	7	23	7	16	5	1	6
4	8	20	8	8	7	24	10	18	7
11	7	7	7	21	5	11	8	17	6
23	4	16	11	0	8	3	9	14	9
5	9	12	9	3	9	5	7	4	12
23	5	5	10	15	7	3	4	23	6
9	9	21	7	23	9	6	9	19	12
9	8	18	10	11	5	12	8	13	8

Tabela A.14: Dados para 40 navios (II)

Apêndice B

Algoritmo da Heurística

```
warning off
clear
arquivo = input('Arquivo de Saida ? ');
fid = fopen(arquivo,'w');
p = [ ];
e = [ ];
w = p + e;
[w,I] = sort(w);
n = ;
m = 2;
L = [0, 0];
berco1 = [ ];
berco2 = [ ];
for i = 1:n
— if L(1) < L(2)
—— berco1 = [ berco1 I(i) ];
—— L(1) =L(1) + p(I(i));
— else
—— berco2 = [ berco2 I(i) ];
—— L(2) = L(2) + p(I(i));
— end
end
```

```

t = [ ];
n1 = length(berco1);
n2 = length(berco2);
tempo1 = 0;
fprintf( fid,' Alocacao no Berco 1 ');
fprintf( fid,' Navio Chegada Inicio Processamento ');
for i = 1:n1
— tempo1 = max( tempo1, e(berco1(i)) );
— t( berco1(i) ) = tempo1;
— fprintf( fid,berco1(i),e(berco1(i)),tempo1,p(berco1(i)));
— tempo1 = tempo1 + p( berco1(i) );
end
tempo2 = 0;
fprintf( fid,' Alocacao no Berco 2 ');
fprintf( fid,' Navio Chegada Inicio Processamento ');
for i = 1:n2
— tempo2 = max( tempo2, c(berco2(i)) );
— t( berco2(i) ) = tempo2;
— fprintf( fid,berco2(i),e(berco2(i)),tempo2,p(berco2(i)));
— tempo2 = tempo2 + p( berco2(i) );
end
fprintf( fid,' Tempo final de processamento no Berco 1:
',tempo1);
fprintf( fid,' Tempo final de processamento no Berco 2:
',tempo2);
espera = t - e;
espera total = sum(espera); fprintf( fid,' Espera total :
',espera total);

```