

Universidade Federal do Maranhão - UFMA  
Centro de Ciências Exatas e Tecnológicas  
Curso de Ciência da Computação

**Análise do modelo matemático proposto para o  
Problema de Alocação de Berços em Portos  
Graneleiros com Restrições de Estoque e  
Condições Favoráveis de Maré**

**Antônio de Sousa Mourão Sobrinho**

**São Luís-MA, Janeiro de 2018**



Antônio de Sousa Mourão Sobrinho

**Análise do modelo matemático proposto para o Problema  
de Alocação de Berços em Portos Graneleiros com  
Restrições de Estoque e Condições Favoráveis de Maré**

**Monografia** apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal do Maranhão – UFMA

Centro de Ciências Exatas e Tecnológicas

Curso de Ciência da Computação

Orientador: Alexandre César Muniz de Oliveira

Coorientador: Francisco Glaubos Nunes Climaco

São Luís-MA

Janeiro de 2018

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).  
Núcleo Integrado de Bibliotecas/UFMA

Sobrinho, Antônio.

Análise do modelo matemático proposto para o Problema de Alocação de Berços em Portos Graneleiros com Restrições de Estoque e Condições Favoráveis de Maré} \autor{Antônio de Sousa Mourão Sobrinho / Antônio Sobrinho. - 2018.  
48 f.

Coorientador(a): Francisco Climaco.

Orientador(a): Alexandre Muniz.

Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, Universidade Federal do Maranhão, 2018.

1. Controle de estoque. 2. Gurobi. 3. Portos com restrições de maré. 4. Problema de alocação de berços. I. Climaco, Francisco. II. Muniz, Alexandre. III. Título.

Antônio de Sousa Mourão Sobrinho

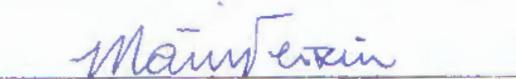
**Análise do modelo matemático proposto para o Problema de Alocação de Berços em Portos Graneleiros com Restrições de Estoque e Condições Favoráveis de Maré**

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. São Luís-MA, 22 de janeiro de 2018

  
Alexandre César Muniz de Oliveira  
Orientador

  
Francisco Glaubos Nunes Climaco  
Co-Orientador

  
Mário Antônio Meireles Teixeira  
Convidado

  
Samyr Bêliche Vale  
Convidado

São Luís-MA  
Janeiro de 2018

# Agradecimentos

Agradeço a Deus por ter me dado conhecimento para desbravar este trabalho e todos os outros.

Agradeço a minha tia, Ana Clara, a minha avó, Zelinda, e minha mãe, Luzimar, por serem mulheres fortes que me deram amor e por terem me ensinado a ser um homem honrado, justo e virtuoso.

Agradeço a minha namorada, Fernanda Lima Mota, por ter me acompanhado nesse caminho, bem como me apoiado e por me fazer querer ser alguém melhor todos os dias.

Agradeço aos professores Alexandre César e Francisco Glaubos por terem me fornecido conhecimento para realização deste trabalho, e em especial ao professor Alexandre que me orienta desde a época do PET em 2012.1.

Agradeço a todos os meus amigos da UFMA, que embora seja difícil citar todos, sempre estavam se fizeram presentes, seja para compartilhar conhecimento, alegrias, vitórias ou mesmo por diversão.

Agradeço ao meu amigo Moisés Rocha, o qual conheci no LACMOR e me ajudou bastante quando precisava saber mais sobre o problema estudado nesse trabalho.

*“Oh! Quão bom e quando suave  
é que os irmãos vivam em união!...”  
(Salmo 133)*



# Resumo

O Problema de Alocação de Berços (PAB) é alvo de muitos estudos em Pesquisa Operacional e é definido como o planejamento de quando e onde(berço) o navio irá atracar afim de minimizar o custos para o porto. Um caso especial é o Problema de Alocação de Berços em Portos Graneleiros com Restrições de Estoque e Condições Favoráveis de Maré, observável nos complexo portuário de São Luís, Maranhão e abordado nesse trabalho. Fazendo uso dos solvers Gurobi e CPLEX para a implementação do modelo matemático do problema, este trabalho analisa o desempenho dos dois solvers em instâncias da literatura e a complexidade do modelo a partir da implementação no Gurobi. O Gurobi tem melhor desempenho em três das 15 instâncias encontradas na literatura, em duas, o presolve realizado pelo Gurobi consegue reduzir o problema mais que o CPLEX. A partir das novas instâncias é possível observar que a quantidade de navios influencia mais no tempo que a mesma será respondida do que a quantidade de berços. Realizando uma regressão polinomial quando a quantidade de berços é constante, é encontrado uma função que comprova o comportamento polinomial do tempo em relação ao crescimento do número de navios.

**Palavras-chaves:** Problema de alocação de berços, Portos com restrições de maré, Controle de estoque, Gurobi.



# Abstract

The Berth Allocation Problem (BAP) is the target of studies on Operational Research and is defined as when and where (berth) the ship will dock in order to minimize costs to the port. A special case is the Berth Allocation Problem in Tidal Builk ports with Sotck level conditions, observable in the port complex of São Luís, Maranhão and approached in this work. Using Gurobi and CPLEX solvers for the implementation of the mathematical model of the problem, this work analyzes the performance of the two solvers in instances of the literature and the complexity of the model from the implementation in Gurobi. The Gurobi performs better in three of the 15 instances found in the literature, in two, the presolve carried out by Gurobi manages to reduce the problem more than the CPLEX. From the new instances it is possible to observe that the quantity of ships influences more in the time that the same will be answered than the quantity of cots. By performing a polynomial regression when the number of berths is constant, a function is found which proves polynomial behavior of the time in relation to the growth of the number of ships.

**Keywords:** Berths allocation problem, Tide restricted ports, Stock control, Gurobi.



# Lista de ilustrações

Figura 1 – Porto de Yangshan em 23 de abril de 2017. Fonte: XINHUANET . . . .	1
Figura 2 – Porto de Santos. Fonte: G1 globo SP . . . . .	1
Figura 3 – Porto do Itaqui. Fonte: O Imparcial . . . . .	2
Figura 4 – Tipos de berços. Fonte: (RODRIGUES et al., 2015) . . . . .	4
Figura 5 – Tempo de uma maré. Fonte: (BARROS, 2010) . . . . .	5
Figura 6 – Construção da árvore co branch & bound . . . . .	10
Figura 7 – Exemplo da estrutura de instância. Fonte: Próprio autor . . . . .	17
Figura 8 – 1 berço. Fonte: Próprio autor . . . . .	24
Figura 9 – 2 berços. Fonte: Próprio autor . . . . .	24
Figura 10 – 3 berços. Fonte: Próprio autor . . . . .	25
Figura 11 – 4 berços. Fonte: Próprio autor . . . . .	25
Figura 12 – 5 berços. Fonte: Próprio autor . . . . .	25
Figura 13 – 6 berços. Fonte: Próprio autor . . . . .	25



# Lista de tabelas

Tabela 1 – Modelo tabale experimento $2^2$ . . . . .	11
Tabela 2 – Resultados . . . . .	18
Tabela 3 – Tempo de execução em segundos . . . . .	19
Tabela 4 – Configurações iniciais das instâncias . . . . .	20
Tabela 5 – Presolve . . . . .	20
Tabela 6 – Média dos tempos presolve (segundos) . . . . .	21
Tabela 7 – Tempo médio de execução das novas instâncias (segundos) . . . . .	22
Tabela 8 – Tempo de execução dos testes usados para o cálculo da inflências . . . . .	22
Tabela 9 – Efeitos dos fatores sobre o tempo médio . . . . .	23
Tabela 10 – Soma dos quadrados . . . . .	23
Tabela 11 – Influência de cada fator . . . . .	23



# Lista de abreviaturas e siglas

API	Application Programming Interface
ANTAQ	Agência Nacional de Transportes Aquaviários
BAPTBS	Berh Allocation Problem in Tidal Bulk porti with Sotck level conditions
PAB	Problema de Alocação de Berços
PL	Programação Linear
PLI	Programação Linear Inteira
PI	Programação Inteira
SSA	Sum of Squares of A
SSB	Sum of Squares of B
SSAB	Sum of Squares of AB
SST	Sum of Squares Total
TEU	Twenty-foot Equivalent Units



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	Motivação	3
1.2	Objetivo	3
1.3	Organização do trabalho	3
<b>2</b>	<b>PROBLEMA DE ALOCAÇÃO DE BERÇOS</b>	<b>4</b>
2.1	Variações do PAB	4
2.2	Cenário utilizado	5
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>7</b>
3.1	Programação Linear	7
3.1.1	Exemplo de programação linear	7
3.1.1.1	Função objetivo	8
3.1.1.2	Restrições	8
3.1.2	Programação Inteira	8
3.1.3	Programação Inteira Mista	8
3.1.4	Presolve	8
3.1.4.1	Exemplos de presolve	9
3.1.5	Relaxação Linear	9
3.1.5.1	Exemplo	9
3.1.6	Branch & Bound	10
3.2	Influência dos fatores sobre um experimento	11
3.2.1	Experimento $2^2$	11
<b>4</b>	<b>MODELO</b>	<b>13</b>
4.1	Dados de entrada	13
4.2	Variáveis de decisão	14
4.3	Função objetivo	14
4.4	Restrições	14
<b>5</b>	<b>RESULTADOS COMPUTACIONAIS</b>	<b>16</b>
5.1	Instâncias	16
5.2	Experimentos	18
5.2.1	Comportamento do tempo de respoa do PAB	22
<b>6</b>	<b>CONCLUSÃO</b>	<b>26</b>
6.1	Trabalhos futuros	26

**REFERÊNCIAS** ..... 27

# 1 Introdução

O comércio entre países ainda tem como pilar o transporte marítimo, 80% (oitenta por cento) do transporte de mercadorias é feito por vias marítimas (BUHRKAL et al., 2011). Ao chegar em um terminal portuário, o navio deverá atracar em um espaço denominado berço, no qual poderá carregar e descarregar a mercadoria. Devido ao grande fluxo de navios e a não existência de berços para todos, é comum encontrar em regiões próximas ao porto navios que desejam atracar.

Em 2010, o porto de Yangshan em Shanghai (Figura 1) foi considerado o maior porto de contêineres do mundo, com 30 milhões de TEUs (unidade equivalente a um contêiner de 20 pés), posto que mantém até os dias atuais. O porto de Yangshan comercializa com mais de 500 portos de 214 países e regiões (XINHUANET, 2017).



Figura 1 – Porto de Yangshan em 23 de abril de 2017. Fonte: XINHUANET



Figura 2 – Porto de Santos. Fonte: G1 globo SP

O anexo da Resolução 2969 da Agência Nacional de Transportes Aquaviários (ANTAQ), informa que no Brasil existem 235 instalações portuárias, públicas ou privadas, de linhas de navegação marítima ou fluvial.

O porto de Santos (Figura 2) é considerado o principal porto do Brasil (SIQUEIRA, 2016), movimentando mais de US\$60 bilhões de dólares por ano. Entre janeiro e outubro

de 2017 passaram por esse porto mais de 100 milhões de toneladas de produtos, um crescimento de 11,65% em relação ao mesmo período de 2016 (SANTOS, 2017). Já entre janeiro e novembro de 2017, o porto de Santos foi responsável por 84,8% da exportação do café brasileiro (ATRIBUNA, 2017).

No Estado do Maranhão há um complexo portuário que se destaca pela movimentação de granéis sólidos, sendo composto pelo porto do Itaqui, Terminal Marítimo da Ponta da Madeira e o Porto da Alumar.

O porto do Itaqui responde por quase 17 milhões de toneladas de produtos, principalmente granéis sólidos derivados de minerais, assim como de vegetais, e, ainda, líquidos derivados do petróleo (ITAQUI, 2017).



Figura 3 – Porto do Itaqui. Fonte: O Imparcial

Com a grande movimentação nos portos e quantidade insuficiente de berços para atracar todos os navios, é normal a formação de filas de espera, como exemplo de tal fato, tem-se que em 21 de dezembro de 2017, às 17hs46min (horário de Brasília), no Porto de Santos, haviam 13 (treze) navios atracados em terminais esperando por um berço para poder carregar ou descarregar, ressaltando-se que ainda eram esperados mais de 24 (vinte e quatro) navios no referido porto no fatídico dia (SANTOS, 2017).

Em decorrência do grande fluxo de navios, faz-se necessário o planejamento de quando e onde o navio irá atracar, frisando-se que um mal planejamento pode acarretar em multa a ser paga pelo porto (*demurrage*) e, também, perda de dinheiro pela menor quantidade de navios atendidos em uma janela de tempo, em contra partida, o bom planejamento resulta em um maior aproveitamento dos berços e lucro para o porto.

Como elucidação para as adversidades acima mencionadas, a Pesquisa Operacional e a Tecnologia da Informação têm se mostrado grandes aliadas no planejamento de operações em terminais portuários pelo mundo, tornando-se imprescindíveis para o aumento de produtividade e redução de custos (MENEZES, 2016).

## 1.1 Motivação

Um berço define um local específico ao longo do cais disponível para carregar ou descarregar um navio por vez. Em portos influenciados por grandes variações das marés, costumam haver sérias restrições à movimentação de navios durante os períodos de baixa mar (BARROS et al., 2011).

O Problema de Alocação de Berços, em Portos Graneleiros, com Restrições de Estoque e Condições Favoráveis de Maré, tem sido estudado desde 2009 como um problema específico dos terminais portuários de São Luís.

Em Barros et al. (2011) e posteriormente Barros (2017), foram propostos modelos de programação linear inteiros para representar o *Berth Allocation Problem in Tidal Bulk ports with Stock level conditions - BAPTBS*.

Recentemente, *solvers* de alto desempenho, como CPLEX (ILOG, Inc, 2006) e Gurobi (GUROBI, 2017), têm avançado muito em performance no que diz respeito a problemas de otimização combinatória, permitindo resolver instâncias que até então eram exclusividade de meta heurísticas bem elaboradas (BARROS, 2010).

## 1.2 Objetivo

Este trabalho tem como objetivo principal avaliar a complexidade do modelo matemático proposto para o Problema de Alocação de Berços em Portos Graneleiros com Restrições de Estoque e Condições Favoráveis de Maré (BARROS et al., 2011) a partir de sua implementação utilizando a API Gurobi (GUROBI, 2017).

Os experimentos permitem comparar o desempenho específico dessa API com o *solver* CPLEX(ILOG, Inc, 2006), bem como indicar direções para a geração de novas instâncias do BAPTBS que sejam desafiadoras para as novas plataformas de otimização de alto desempenho existentes atualmente.

## 1.3 Organização do trabalho

O trabalho está dividido em cinco capítulos, sendo que neste capítulo foi apresentado a contextualização do problema, como também o objetivo de fazer este trabalho. No Capítulo 2 será apresentado o problema de um modo mais genérico, mostrando as variações do mesmo. Já no Capítulo 3, serão apresentadas técnicas de programação inteira utilizadas pelos *solvers* analisados. O modelo trabalhado e todas as suas particularidades serão apresentados no Capítulo 4. Os resultados computacionais realizados serão apresentados no Capítulo 5. Por fim, o Capítulo 6 destacará as principais conclusões tiradas dos experimentos e, ainda, como este trabalho pode ser continuado.

## 2 Problema de alocação de berços

O Problema de Alocação de Berços (PAB) é definido como o planejamento de quando e em qual berço o navio irá atracar para realizar o carregamento ou descarregamento de granéis (CORDEAU et al., 2005). Sua criação dar-se-á da seguinte forma: será posicionada uma fila de espera caso não haja berço para que o navio atraque no cais (FERNANDES, 2001); logo, o planejamento deverá ser feito de modo a minimizar o tempo de atendimento geral, os custos operacionais e as possíveis multas a serem pagas pelo porto.

### 2.1 Variações do PAB

Vários modelos do PAB já foram criados. BARROS (2010) considera o tempo de chegada do navio, a quantidade da carga e o tipo de berço que o porto possui, Menezes (2016) considera também os equipamentos de transporte do produto pelo porto, as rotas que o produto pode tomar desde a chegada no porto até o navio, e o período de chegada do produto no porto, o que torna o problema bem maior.

Os berços podem ser classificados quanto ao modo de atracação (Figura 4) e quanto a vazão.

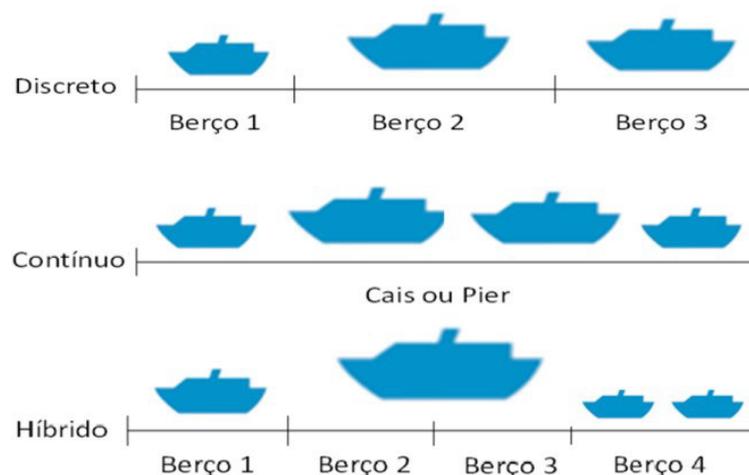


Figura 4 – Tipos de berços. Fonte: (RODRIGUES et al., 2015)

- Modo de atracação
  - Discreto: Cada berço comporta apenas um navio;
  - Contínuo: Não há divisão entre os berços, o desafio é alocar os navios e melhor jeito possível;

- Híbrido: O espaço de atracação é dividido em berços, mas com a possibilidade de mais de um navio em um berço ou um navio em vários berços;
- Vazão
  - Homogêneos: Todos os berços possuem a mesma velocidade de vazão;
  - Heterogêneos: Os berços podem possuir velocidades de vazão diferentes;

O PAB ainda pode ser tratado de forma estática ou dinâmica. O PAB estático considera apenas os navios que já chegaram e os que já estão sendo atendidos, enquanto que o PAB dinâmico considera também os navios cuja chegada já é prevista (CORDEAU et al., 2005). Os tempos de chegada, atracagem e saída dos navios também podem ser representados de forma contínua ou ser divididos em janelas de tempo (CORDEAU et al., 2005). BARROS (2010) utiliza as marés como janela de tempo, sendo uma maré o tempo de maré alta, preamar, mais o período de maré baixa, baixamar. A Figura 5 representa o tempo de maré. Em São Luís um período de maré (preamar + baixamar) dura em média 12 horas (APOLO11.COM, 2017).

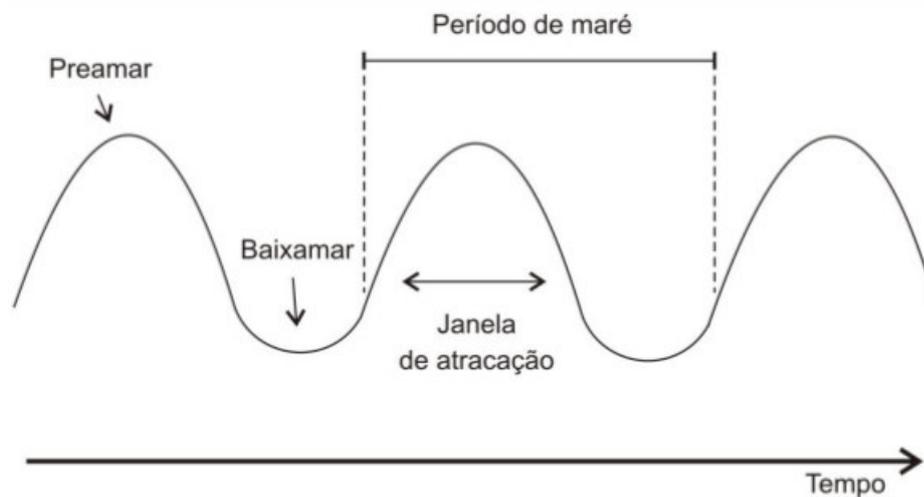


Figura 5 – Tempo de uma maré. Fonte: (BARROS, 2010)

## 2.2 Cenário utilizado

O modelo utilizado neste trabalho foi o formulado por Barros (2017) que tem como base o complexo portuário de São Luís com as seguintes características:

- Os berços devem ser tratados de forma discreta, cada um comportando apenas um navio por vez;
- Os berços possuem velocidade de carga e descarga diferentes, sendo tratados de forma heterogênea;

- O tempo de atendimento e chegada serão em janelas de tempo(tempo de maré), será considerado que um navio chega e é atendido no início da janela e termina o atendimento no final de uma janela nunca podendo sair ou entrar no berço no meio da janela de tempo;
- Serão considerados, além dos navios em atendimento e os na fila de espera, os navios que já possuem chegada prevista
- O navio deverá terminar o atendimento no mesmo berço que começou, não sendo permitido a troca de berço durante o atendimento;
- O estoque dos produtos devem ser mantidos em níveis seguros para que um navio não fique parado esperando por um produto não disponível no porto;

Esse cenário em especial é nomeada Berth Allocation Problem in Tidal Bulk ports with Stock level conditions - BAPTBS

## 3 Fundamentação Teórica

### 3.1 Programação Linear

A programação linear (PL) é definida como um problema que procura otimizar (maximizar ou minimizar) uma função linear, denominada de função objetivo, respeitando restrições de igualdade ou desigualdade que também devem ser lineares (FERGUSON, 2000).

Por volta de 1947, George B. Dantizing formulou o conceito de programação linear enquanto trabalha como orientador matemático na Força Aérea dos Estados Unidos (BAZARAA; JARVIS; SHERALI, 2010).

Um modelo matemático tem o objetivo de descrever, sem erros, algum problema real. Alguns problemas reais se tornaram clássicos no estudo da programação linear, como por exemplo: alocação de recursos na indústria; criação de fórmulas no ramo alimentício, rações e adubos; decidir sobre onde investir; construir a tabela de horários dos funcionários e, criação de ligas metálicas (SILVA, 2016).

A estrutura da programação é mostrada a seguir:

$$MIM/MAX e^t x \quad (3.1)$$

$$Ax = b \quad (3.2)$$

$$k \leq x \leq z \quad (3.3)$$

A equação 3.1 é a função objetivo, na qual se busca maximizar ou minimizar o seu valor, enquanto que as equações 3.2 e 3.3 representam as restrições que as variáveis de decisão estão sujeitas, sendo a 3.3 representando as restrições de limite superior e limite inferior das variáveis.

#### 3.1.1 Exemplo de programação linear

Uma empresa possui dois produtos,  $x$  e  $y$ , sendo que a empresa tem R\$ 7,00 de lucro com o produto  $x$ , e, R\$ 5,00 de lucro com o produto  $y$ , gasta R\$ 2,00 para fazer o produto  $x$  e R\$ 1,50 para fazer o produto  $y$  e tem R\$ 1000,00 no total para a produção.

### 3.1.1.1 Função objetivo

$$MAX 7x + 5y \quad (3.4)$$

### 3.1.1.2 Restrições

$$2x + 1.5y \leq 1000 \quad (3.5)$$

$$x \geq 0 \quad (3.6)$$

$$y \geq 0 \quad (3.7)$$

Neste exemplo as variáveis de decisão são  $x$  e  $y$  representando a quantidade do produto homônimo e o objetivo é maximizar a quantidade de  $x$  vezes o lucro obtido com o mesmo mais a quantidade de  $y$  vezes o lucro obtido com  $y$ . A restrição 3.2 gasto mais do que a empresa possui. As restrições 3.3 e 3.4 garantem que não seja produzido produtos em quantidade negativa, o que é impossível no cenário real.

## 3.1.2 Programação Inteira

A programação inteira (PI), também chamada de programação linear inteira (PLI), é uma sub-classe da programação linear onde todas ou algumas das variáveis de decisão pertencem ao conjunto dos inteiros. Apesar do espaço de busca ser menor do que se as variáveis fossem contínuas, a programação linear inteira se mostrou um problema NP-difícil (SCHRIJVER, 1998). Um tipo especial de PI é se possuir variáveis binárias, este tipo de PI também é classificado como NP-difícil.

## 3.1.3 Programação Inteira Mista

A programação inteira mista (Mixed-Integer Programming - MIP), engloba problemas onde há variáveis inteiras e permite que outras variáveis sejam não inteiras.

## 3.1.4 Presolve

Presolve é um conjunto de técnicas utilizadas para reduzir um problema de programação linear. A dimensionalidade de  $A$  na equação 3.2 é importante para definir o tamanho de um problema em PL, porque apenas os  $A$ s não zeros são resolvidos e o problema cresce com o tamanho dos não zeros. Uma técnica de presolve procura reduzir o problema reduzindo também o número de não zeros em  $A$ , nunca criando novos (ANDERSEN; ANDERSEN, 1995).

### 3.1.4.1 Exemplos de presolve

Há algum problema que é sujeito as restrições a seguir:

$$x + y + z \geq 41 \quad (3.8)$$

$$x \leq 11, x \in \mathbb{Z} \quad (3.9)$$

$$y \leq 13, y \in \mathbb{Z} \quad (3.10)$$

$$z \leq 17, z \in \mathbb{Z} \quad (3.11)$$

Com isso, a única solução é  $x = 11$ ,  $y = 13$  e  $z = 17$ . Assim essas três variáveis podem ser retiradas da formulação, como também as 4 restrições que dependem apenas delas, tornando o problema mais fácil de ser solucionado.

O presolver ainda pode retirar restrições redundantes como no caso das equações 3.12 e 3.13 onde a primeira pode ser removida.

$$4x + 4y \geq 16 \quad (3.12)$$

$$x + y \geq 4 \quad (3.13)$$

### 3.1.5 Relaxação Linear

A técnica de relaxação linear consiste em retirar a restrição da PI que garante que as variáveis de decisão sejam inteiras e transformar-lás em contínuas, assim o problema ficará mais fácil de ser resolvido e o simplex conseguirá achar um resultado muito rápido. A solução de um problema relaxado será sempre melhor ou igual a solução do problema inteiro.

#### 3.1.5.1 Exemplo

$$\text{Max } 3x + 5y \quad (3.14)$$

$$6x + 8y \leq 48 \quad (3.15)$$

$$4x + 10y \leq 40 \quad (3.16)$$

$$x, y \geq 0 \quad (3.17)$$

O modelo descrito nas equações 3.14 até 3.17 não possuem a restrição para garantir que  $x$  e  $y$  sejam inteiros, sendo assim, métodos como o de gráfico facilmente resolvem esse modelo. Nesse caso, o ótimo será 25,68 com  $x = 5,71$  e  $y = 1,71$ , este valor com certeza é maior do que se houvesse a restrição permitindo apenas inteiros.

### 3.1.6 Branch & Bound

O algoritmo de Branch & Bound utiliza as variáveis relaxadas e vai criando uma árvore de subproblemas do modelo original. A ramificação é parada quando:

- Uma solução é composta somente por variáveis inteiras, a solução é chamada de incumbente;
- A solução relaxada é pior que a solução incumbente atual;
- A solução composta somente por soluções inteiras é pior que a solução incumbente atual;

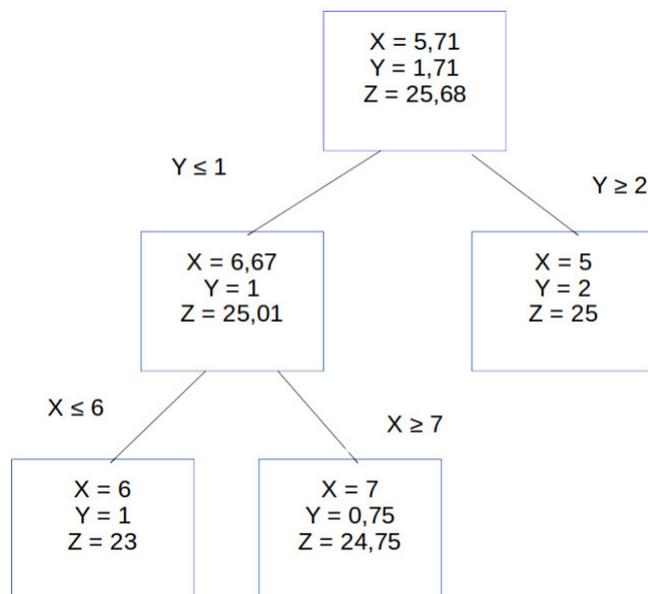


Figura 6 – Construção da árvore co branch & bound

A continuação do problema pode ser visto na figura 6. Como as duas variáveis de decisão não são inteiras, é escolhido qualquer umas para adicionar uma nova restrição que inviabilize a solução atual, neste caso o  $y$  foi escolhido, sendo criado dois subproblemas, um com  $y \leq 1$  e outro com  $y \geq 2$ . O segundo subproblema encontra uma solução com variáveis

inteiras, este será o valor incumbente atual, logo não precisa mais ser dividido em novos subproblemas. O problema com  $y \leq 1$  encontra uma solução melhor que a incumbente atual, ele precisa ser dividido e como apenas o  $x$  não é inteiro ele é escolhido para ser alvo da nova restrição. Com a adição de um ramo com restrição de  $x \leq 6$  é encontrado uma solução de variáveis inteiras mas menor que a incumbente atual. A solução encontrada com a adição do ramo com a restrição  $x \geq 7$  contém uma variável não inteira ( $y = 0,75$ ), porém, como o valor da função objetivo é pior que o valor incumbente atual, não há a necessidade da adição de novas restrições, pois nenhum valor a partir de agora será maior que os já encontrados.

## 3.2 Influência dos fatores sobre um experimento

Um experimento pode ser afetado por  $k$  fatores, estes possuindo  $n$  níveis. No caso do BAPTBS, quantidade de berços, navios, marés, consumo e estoque inicial pode ser considerados fatores que afetam, e o valores que eles assumem são seus níveis. JAIN explica que um projeto fatorial completo (que considera todo conjunto  $k$ ) com um número grande de níveis pode não ser o melhor uso do esforço disponível, por isso é aconselhável escolher uma pequena quantidade de fatores que realmente afetam o experimento.

### 3.2.1 Experimento $2^2$

Um experimento  $2^2$  é um caso de  $2^k$  onde  $k = 2$ , ou seja, são dois fatores com dois níveis cada. Uma tabela é montada nos moldes da tabela 1, onde há o caso base (O), no segundo experimento é modificado o fator A, no terceiro o fator B e no quarto tanto A quanto B são modificados e é tirado a resposta (Y) de cada experimento.

Tabela 1 – Modelo tabale experimento  $2^2$

Experimento	Fator A	Fator B	Y
1(O)	-1	-1	$y_1$
2(A)	1	-1	$y_2$
3(B)	-1	1	$y_3$
4(AB)	1	1	$y_4$

As respostas observadas podem ser definidas por:

$$y = q_O + q_A x_A + q_B x_B + q_{AB} x_{AB} \quad (3.18)$$

Onde  $x \in (1, -1)$  de acordo com a tabela 1, e os  $q_s$  são chamados de efeitos. Assim, podemos obter os  $y$  por:

$$y_1 = q_O - q_A - q_B + q_{AB} \quad (3.19)$$

$$y_2 = q_O + q_A - q_B - q_{AB} \quad (3.20)$$

$$y_3 = q_O - q_A + q_B - q_{AB} \quad (3.21)$$

$$y_4 = q_O + q_A + q_B + q_{AB} \quad (3.22)$$

Isolando o  $q_s$  teremos:

$$q_O = \frac{1}{4}(y_1 + y_2 + y_3 + y_4) \quad (3.23)$$

$$q_A = \frac{1}{4}(-y_1 + y_2 - y_3 + y_4) \quad (3.24)$$

$$q_B = \frac{1}{4}(-y_1 - y_2 + y_3 + y_4) \quad (3.25)$$

$$q_{AB} = \frac{1}{4}(y_1 - y_2 - y_3 + y_4) \quad (3.26)$$

Com os valores de  $q_s$  calculados, é possível obter a soma dos quadrados total, do inglês, Sum of Squares Total (SST), com mostra na equação 3.28.

$$SST = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2 \quad (3.27)$$

ou:

$$SST = SSA + SSB + SSAB \quad (3.28)$$

E por fim, a influências que cada fator tem sobre Y é obtido dividindo a soma dos quadrados parcial pelo total, como visto nas equações 3.29, 3.30 e 3.31.

$$Influência de A = \frac{SSA}{SST} \quad (3.29)$$

$$Influência de B = \frac{SSB}{SST} \quad (3.30)$$

$$Influência de AB = \frac{SSAB}{SST} \quad (3.31)$$

## 4 Modelo

O modelo utilizado nesse trabalho foi proposto por Barros (2017). Representa um porto com  $L$  berços heterogêneos, leva em consideração os  $N$  navios que a chegada já é prevista durante o tempo de  $M$  marés, cada navio possui o tempo de chegada  $a$  e será carregado com um produto  $k$  pertencente ao conjunto  $K$ .

Barros (2017) considerou o que o porto modelado transporta apenas granel, assim não há problema de alocação de contêiner. O tempo de atendimento de um navio é a relação entre a quantidade de carga pela vazão do berço arredondado para cima, pois os navios não podem entrar ou sair do berço durante a janela de tempo. A vazão do berço será a mesma para todas as cargas.

### 4.1 Dados de entrada

- $N$ : quantidade de navios;
- $M$ : quantidade de marés;
- $L$ : quantidade de berços;
- $K$ : conjunto das cargas;
- $a_i$ : maré de chegada do navio  $i$ ;
- $v_l$ : vazão do berço  $l$ ;
- $e_k$ : estoque inicial do produto  $k$  pertencente ao conjunto  $K$ ;
- $c_k$ : produção de  $k$  pertencente ao conjunto  $K$ ;
- $q_{ik}$ : quantidade do produto  $k$  transportado pelo navio  $i$ ;

A partir da entrada é obtido o tempo, em marés, que um berço  $l$  consegue atender o navio  $i$ , esse valor é representado pelo dado  $h_{il}$  e é calculado por:

$$h_{il} = \left\lceil \frac{\sum_{k=1}^{|K|} q_{ik}}{v_l} \right\rceil \quad (4.1)$$

## 4.2 Variáveis de decisão

- $y_{ijl}$ : Representa se o navio  $i$  atracará no berço  $l$  na maré  $j$ . 1 se sim e 0 se não;

A variável  $y_{ijl}$  é a decisão de quando o navio irá atracar ( $j$ ) e em qual berço o navio irá atracar ( $l$ ). Como as variáveis de decisão são todas binárias então o problema será NP-difícil, como mostrado no Capítulo 3

## 4.3 Função objetivo

$$\min \sum_{i=1}^{|N|} \sum_{j=1}^{|M|} \sum_{l=1}^{|L|} (j + h_{il} - a_i) y_{ijl} \quad (4.2)$$

A função objetivo busca minimizar a soma do tempo, em marés, de atendimento total. O tempo de atendimento total é a soma do tempo na fila de espera, que seria a que o navio atracou menos a maré de chegada ( $j - a_i$ ), mais o tempo de atendimento no berço escolhido ( $h_{il}$ ).

## 4.4 Restrições

O modelo está sujeito as seguintes restrições:

$$\sum_{j=1}^{a_i-1} \sum_{l=1}^{|L|} y_{ijl} = 0, \forall i \in N \quad (4.3)$$

$$\sum_{j=a_i}^{|M|} \sum_{l=1}^{|L|} y_{ijl} = 1, \forall i \in N \quad (4.4)$$

$$\sum_{\substack{n=1 \\ n \neq i}}^{|N|} \sum_{\substack{m=j \\ m \leq |M|}}^{j+h_{ij}-1} y_{nml} \leq (1 - y_{ijl})|N|, \forall i \in N, \forall j \in M, \forall l \in L \quad (4.5)$$

$$\sum_{i=1}^{|N|} \sum_{l=1}^{|L|} \sum_{z=a_i}^j \frac{\min(j - a_i + 1, h_{il})}{h_{il}} q_{ik} * y_{izb} \leq j * c_k + e_k, \forall j \in M, \forall b \in L, \forall k \in K \quad (4.6)$$

A restrição 4.3 impede que seja achado uma solução na qual o navio seja atendido em uma maré antes da maré na qual o mesmo chegou, enquanto a restrição 4.4 garante que o começo do atendimento se dará em apenas uma maré a partir da maré de chegada.

A restrição 4.5 impede que algum navio atraque em um berço enquanto o mesmo está atendendo outro navio. Se  $y_{ijl}$  for 1, representando que o navio  $i$  atracou no berço  $l$  na maré  $j$ , nenhum navio  $n$  poderá atracar no mesmo berço  $l$  no período entre  $j$  e  $h_{il} - 1$ .

O controle de estoque é garantido pela restrição 4.6. A quantidade produzida até a maré  $j$  é calculado por  $j * c_k + e_k$ , ou seja, o estoque inicial mais o produzido até aquela maré. O consumo dos navios até a maré  $j$  é calculado pela porcentagem do atendimento vezes a quantidade de carga para o navio vezes se ele já foi atendido. A função  $mim$  serve para que quando  $j$  for após o atendimento do navio seja usado o valor de  $h_{il}$  para que a função seja  $q_{ik} * y_{izb}$ , representando o atendimento total do navio.

## 5 Resultados computacionais

Para a realização dos testes deste trabalho foi utilizado a API Gurobi ([GUROBI, 2017](#)) na versão 7.5 e o CPLEX Studio versão 12.8. O Gurobi foi escolhido por ser uma alternativa ao principal solver usado comercialmente e no meio acadêmico, o CPLEX. Enquanto a versão individual do CPLEX custa aproximadamente \$21.000,00 (vinte e um mil dólares) ([IBM, 2017](#)), a versão individual do Gurobi custa \$14.000,00 (quatorze mil dólares) ([GUROBI, 2017](#)), uma diferença de quase \$7.000,00 (sete mil dólares), o que é uma quantia significativa caso o Gurobi consiga obter a mesma performance do CPLEX. Foram usadas as licenças de estudante do Gurobi e o CPLEX Studio.

Todos os experimentos foram realizados na mesma máquina, com um Intel(R) Core(TM) i7-2630QM CPU 2.00GHz, 4GB de memória RAM DDR3 e Ubuntu Mate 16.04.

### 5.1 Instâncias

As instâncias utilizadas para validação da implementação no Gurobi foram as mesmas utilizadas por [BARROS \(2010\)](#) e [Barros \(2017\)](#) com pequenas modificações na estrutura, deixando-as mais compactas sem alterar o significados das informações, por exemplo, se há 3 navios [BARROS \(2010\)](#) coloca "N 1 2 3", na nova estrutura ficaria só "N 3". Essas instâncias representam um cenário real de um porto de graneis com características que fazem com que cada restrição do modelo tenha influência significativa na solução. As instâncias variam entre 10 e 30 navios( $N$ ), 15 e 80 marés( $M$ ) e entre 2 e 4 berços( $L$ ) e são nomeadas com base nesses atributos ( $N.M.L$ ). O tamanho da instância é dado por  $N * M * L$ , mas como as outras características, como estoque inicial, também são relevantes, instâncias grandes não necessariamente significam que são mais difíceis. A figura 7 ilustra como as instâncias utilizadas são apresentadas.

Todas instâncias possuem o valor total de navios, definido por  $N$ , a quantidade de marés, definido por  $M$ , o conjunto de graneis, definido por  $K$  e a quantidade de berços, definido por  $L$ , em  $v$  temos a velocidade do atendimento de cada berço, em  $a$  temos a maré de chegada de cada navio, em  $e$  é informado o estoque inicial de cada produto, em  $c_k$  temos a produção de cada produto pro maré e a matriz  $q$  informa quanto cada navio ira consumir dos produtos.

```

N 30
M 80
K Alumina Bauxita Manganes Ferro ;
L 2
v
1 5
2 7
a
1 1
2 1
3 3
4 3
5 5
6 7
7 7
8 8
9 8
10 10
11 11
12 11
13 12
14 12
15 14
16 16
17 17
18 17
19 18
20 19
21 19
22 21
23 23
24 24
25 26
26 28
27 28
28 28
29 30
30 30
e
Bauxita 30
Alumina 39
Manganes 61
Ferro 21
ck
Bauxita 2
Alumina 3
Manganes 4
Ferro 2
q Bauxita Alumina Manganes Ferro
1 0 0 37 0
2 41 0 0 0
3 0 16 0 0
4 22 0 0 0
5 0 0 42 0
6 0 0 33 0
7 13 0 0 0
8 0 45 0 0
9 0 0 23 0
10 0 22 0 0
11 0 0 0 25
12 0 0 0 27
13 0 0 34 0
14 0 0 33 0
15 0 0 0 19
16 20 0 0 0
17 0 39 0 0
18 0 20 0 0
19 0 0 0 21
20 0 0 0 24
21 15 0 0 0
22 0 24 0 0
23 0 18 0 0
24 0 0 33 0
25 15 0 0 0
26 0 23 0 0
27 28 0 0 0
28 0 0 0 25
29 0 17 0 0
30 0 0 46 0

```

Figura 7 – Exemplo da estrutura de instância. Fonte: Próprio autor

## 5.2 Experimentos

Os experimentos consistiram na implementação do modelo matemático proposto por Barros (2017), no Gurobi e no CPLEX Studio, bem como a verificação se ambos conseguem atingir o valor ótimo em tempo viável (3600 segundos), e a análise do *logs* fornecidos pelo Gurobi e pelo CPLEX.

Inicialmente, foram realizadas 30 execução com cada uma das 15 instâncias, tanto para o CPLEX quanto para o Gurobi, totalizando 900 execuções. Uma vez que ambos os solvers Gurobi e CPLEX foram capazes de atingir a solução ótima para todas as instâncias (ver Tabela 2), foi-se analisado o tempo computacional gasto para atingir tais soluções.

Tabela 2 – Resultados

	Instância	Gurobi	CPLEX
1	10.15.3	31	31
2	10.15.4	29	29
3	10.20.3	44	44
4	10.25.2	43	43
5	15.20.4	40	40
6	15.25.3	55	55
7	15.30.4	77	77
8	15.35.3	86	86
9	15.40.2	121	121
10	20.40.3	107	107
11	20.40.4	108	108
12	20.55.2	287	287
13	30.70.4	185	185
14	30.75.3	190	190
15	30.80.2	607	607

A Tabela 3 relaciona cada instância com a média, desvio padrão e intervalo de confiança do tempo de execução do Gurobi e do CPLEX, o intervalo foi calculado com nível de confiança de 95% . A partir da Tabela 2, percebe-se que todas as instâncias foram resolvidas em menos de 1 hora e o Gurobi foi mais rápido apenas três instâncias das 15 testadas, 10.15.3, 10.15.4 e 30.80.2. Com base na média dos tempos e no intervalo de confiança das amostras, é possível observar que nenhum tempo é estatisticamente igual, e que quanto maior e mais difícil é a instância, maior é a variação no tempo de execução do Gurobi enquanto que o CPLEX mantém-se mais constante.

Da instância 1 até a instância 7, a diferença de tempo de execução do Gurobi e CPLEX é pequena. A partir da instância 8 essa diferença entre o Gurobi e o CPLEX é ser mais relevante. O Gurobi gasta 61,64 segundos para resolver a instância 30.70.4, quase o triplo do tempo despendido pelo CPLEX (21,16 segundos).

Para ambos os solver, a instância mais difícil de resolver foi a 15 (30.80.2). Enquanto

Tabela 3 – Tempo de execução em segundos

	Instância	Gurobi			CPLEX		
		Média	Desv. Pad.	Int. Conf.	Média	Desv. Pad.	Int. Conf.
1	10.15.3	0,17	0,01	0,001	0,24	0,01	0,006
2	10.15.4	0,16	0,01	0,002	0,23	0,01	0,004
3	10.20.3	0,41	0,01	0,002	0,33	0,02	0,007
4	10.25.2	0,42	0,01	0,001	0,31	0,01	0,003
5	15.20.4	0,46	0,01	0,001	0,38	0,06	0,002
6	15.25.3	0,90	0,01	0,002	0,52	0,01	0,003
7	15.30.4	1,78	0,02	0,007	1,37	0,01	0,007
8	15.35.3	3,72	0,03	0,009	2,01	0,09	0,010
9	15.40.2	3,70	0,04	0,013	3,78	0,06	0,029
10	20.40.3	5,33	0,03	0,008	2,92	0,01	0,006
11	20.40.4	5,14	0,04	0,010	2,99	0,04	0,020
12	20.55.2	20,24	3,15	1,090	15,38	0,05	0,296
13	30.70.4	61,64	5,19	1,390	21,16	0,07	0,037
14	30.75.3	30,59	4,51	1,470	18,51	0,17	0,084
15	30.80.2	92,01	4,59	1,640	1411,74	1,20	1,050

a diferença de tempo nas demais instâncias não foram mais de 1 minuto, na instância 15 o CPLEX resolve, em média, em 1411,74 segundos em média, aproximadamente 24 minutos, enquanto o Gurobi consegue resolver o mesmo problema em 92,01 segundos, um pouco mais de 1 minuto e meio. A diferença de tempo de processamento na instância 15 é de aproximadamente 22 minutos, a maior diferença entre as médias dos experimentos.

Apesar de ambos os solvers serem proprietários e de código fechado, o que torna um pouco difícil de saber como funcionam, o CPLEX Studio e o Gurobi fornecem algumas informações que podem ajudar a entender o como o processo de resolução dos problemas está sendo realizado, e em alguns casos, a entender os motivos de um solver possuir desempenho melhor que outro, para uma certa instância.

A Tabela 4 apresenta as configurações iniciais de cada instância, isto é, a quantidade de restrições aplicadas à respectiva instância, a quantidade de variáveis de decisão do problema e a quantidade de não zeros na matriz  $Ax = b$ . A quantidade de variáveis pode ser obtida pela expressão  $N * M * L$ . Quanto maior for o número de variáveis, maior será a instâncias, pois possuirá um maior número de restrições aplicadas a ela.

Uma análise das Tabelas 3 e 4 confirma que não necessariamente a maior instância é a mais difícil de resolver. A maior instância testada foi a 13 (30.70.4), sujeita a 8740 restrições, possuindo 8400 variáveis de decisão e 1322657 não-zeros. Foi necessário, em média, 61,64 segundos para resolvê-la pelo Gurobi e 21,16 segundos para resolvê-la pelo CPLEX. Já para a instância 15 (30.80.20), com 5180 restrições, 4800 variáveis e 815.128 não-zeros obteve-se tempo médio no Gurobi de 91,01 segundos e 1411,74 segundo no CPLEX.

Antes de resolver qualquer instância, tanto o Gurobi quanto o CPLEX realizam técnicas de pré-processamento (presolve) para reduzir o problema e facilitá-lo. Os métodos de presolve utilizados pelos solvers não são informados mas o resultado do no *log* do solver pode ser vistos na tabela 5. Na Tabela 5, são apresentados os números de variáveis, restrições e não zeros de cada instância testada, após a execução do presolve.

Tabela 4 – Configurações iniciais das instâncias

	Instância	Restrições	Variáveis	Não zeros
1	10.15.3	530	450	14622
2	10.15.4	680	600	19479
3	10.20.3	700	600	26223
4	10.25.2	620	500	20056
5	15.20.4	1310	1200	55492
6	15.25.3	1255	1125	61431
7	15.30.4	1950	1800	138166
8	15.35.3	1745	1575	114342
9	15.40.2	1390	1200	89670
10	20.40.3	2600	2400	213457
11	20.40.4	3400	3200	334029
12	20.55.2	2460	2200	249561
13	30.70.4	8740	8400	1322657
14	30.75.3	7110	6750	937701
15	30.80.2	5180	4800	815128

Tabela 5 – Presolve

	Instância	CPLEX			Gurobi		
		Restrições	Variáveis	Não-zeros	Restrições	Variáveis	Não-zeros
1	10.15.3	280	272	6850	224	278	5153
2	10.15.4	272	260	4971	138	255	2550
3	10.20.3	356	363	11689	395	387	11918
4	10.25.2	264	350	8023	340	321	9490
5	15.20.4	295	584	9389	520	564	12723
6	15.25.3	358	618	15714	590	580	20827
7	15.30.4	868	944	48356	1110	1076	60361
8	15.35.3	1120	1232	78870	1239	1182	77229
9	15.40.2	892	936	59738	953	936	61566
10	20.40.3	1159	1761	104456	1818	1764	134881
11	20.40.4	1125	1988	106705	2281	2224	190780
12	20.55.2	1682	1830	186376	1961	1840	198705
13	30.70.4	4487	4292	580771	6836	6692	1015312
14	30.75.3	3115	5379	521375	5555	5379	723756
15	30.80.2	3518	3938	568573	4164	3946	647791

Das 15 instâncias testadas, em apenas duas o presolve do Gurobi conseguiu reduzir mais o problema, que o presolve do CPLEX, a instância 1 (10.15.3) ficou com 224 restrições,

279 variáveis e 5153 não-zeros no Gurobi e 280 restrições, 272 variáveis e 6850 não-zeros no CPLEX. A instância 2 (10.15.3) ficou com 138 restrições, 255 variáveis e 2550 não-zeros no Gurobi, enquanto que no CPLEX a redução foi para 272 restrições, 260 variáveis e 4971 não-zeros.

Os dados do presolve das instâncias 1 e 2 talvez expliquem o motivo pelo qual o Gurobi conseguiu resolver essas instâncias em menor tempo que o CPLEX, porém o presolve do CPLEX na instância 15 (30.80.2) obteve, aparentemente, melhor desempenho comparado com o presolve do Gurobi. No CPLEX, a instância 15 foi reduzida para 3518 restrições, 3938 variáveis e 568573 não-zeros, enquanto que no Gurobi a instância ficou com 4164 restrições, 3946 variáveis e 647791 não-zeros, sendo assim, maior que a do CPLEX. Mesmo o Gurobi não reduzindo tanto o problema da instância 15, esta foi resolvida muito mais rápida do que no CPLEX, como apresentado na Tabela 3, uma hipótese é que o conjunto de restrições mantidas pelo Gurobi sejam mais fáceis de serem cumpridas do que as restrições mantidas pelo CPLEX, sendo assim, o subproblema do Gurobi é de mais fácil resolução.

Devido a ambos os solvers, Gurobi e CPLEX, serem comerciais e de código fechado, não há como saber os critérios utilizados na etapa de presolve, mas evidências nos levam a acreditar que ambos utilizam técnica diferentes, já que para instâncias iguais, o resultado das reduções foram diferentes.

Tabela 6 – Média dos tempos presolve (segundos)

	Instância	Gurobi	CPLEX
1	10.15.3	0,07	0,01
2	10.15.4	0,08	0,01
3	10.20.3	0,11	0,02
4	10.25.2	0,1	0,02
5	15.20.4	0,22	0,01
6	15.25.3	0,2	0,05
7	15.30.4	0,49	0,6
8	15.35.3	0,51	0,7
9	15.40.2	0,32	0,12
10	20.40.3	0,87	0,73
11	20.40.4	1,27	1
12	20.55.2	1,08	1,47
13	30.70.4	6,12	4,21
14	30.75.3	3,85	5,34
15	30.80.2	3,53	7,07

A Tabela 6 apresenta a média do tempo que o CPLEX e o Gurobi utilizam para realizar a etapa de presolve. Percebe-se que o comportamento quanto ao tempo para realizar o presolve no Gurobi é diferente no CPLEX.

### 5.2.1 Comportamento do tempo de resposta do PAB

Além das 15 instâncias utilizadas no trabalho de [BARROS \(2010\)](#), neste trabalho, são propostas novas instâncias, variando o número de berços (entre 1 e 6) e o número de navios (entre 5 e 30), sendo 5 de cada, classificadas com as letras A, B, C, D e E, totalizando 180 nova instâncias. As instâncias com a mesma letra possuem o mesmo valor de estoque, consumo e seus navios com mesmo índice possuem a mesma maré de chegada e mesma carga, o número de marés é o mesmo para todas as instâncias. Foram colhidos os tempos de execução médio de cada tipo para analisar o comportamento do tempo de resposta do PAB.

A tabela 7 mostra a média dos tempos de execução de cada tipo de instância. As colunas são a quantidade de navios enquanto as linhas informam a quantidade de berços presente na classe. Instâncias com 50 navios foram criadas, mas algumas se mostram inviáveis de serem respondidas em 3600 segundos.

Tabela 7 – Tempo médio de execução das novas instâncias (segundos)

	5	10	15	20	25	30
1	0,37	3,79	15,67	38,91	100,88	159,37
2	1,62	4,5	10,89	22,5	107,93	165,15
3	2,12	5,18	10,83	21,50	63,63	109,12
4	2,37	5,89	11,64	22,98	53,92	310,24
5	2,63	6,92	13,26	39,02	95,54	298,29
6	2,84	9,78	14,6	68,85	148,53	491,86

Para uma análise melhor do comportamento do problema, foi realizado o cálculo da influência da quantidade de berços e navios no tempo médio de execução. Foi escolhido os dois casos extremos, 5 e 30 navios e 1 e 6 berços para realizar o cálculo (Tabela 8).

Tabela 8 – Tempo de execução dos testes usados para o cálculo da influências

Experimento	Navios(A)	Berços(B)	Tempo médio
1(O)	5	1	0,37
2(A)	30	1	159,37
3(B)	5	6	2,84
4(AB)	30	6	491,86

Com os dados dos casos extremos, foi obtido o efeito que a mudança na quantidade de navios (A), berços (B), os dois juntos (AB) e nenhum (O) tem sobre o tempo de execução.

Pelos dados da tabela 9 é possível notar que o fator A (navios) tem efeito maior que o B (berço) ou os dois juntos (AB). Após a obtenção dos efeitos, foi realizado a equação das somas dos quadrados, como mostradado no Capítulo 3, o resultado pode ser visto na tabela 10. A partir da soma dos quadrados, foi possível obter a influência que cada fator

Tabela 9 – Efeitos dos fatores sobre o tempo médio

Fatores	Efeito
0	163,61
A	162,00
B	83,73
AB	82,50

tem sobre o tempo de resposta do BAPTBS. A tabela 11 mostra que o fator A (Navio) tem 65% de influência sobre o tempo em que o PAB será respondido utilizando o Gurobi, influenciando bem mais do que a quantidade de berço ou se modificar os dois ao mesmo tempo.

Tabela 10 – Soma dos quadrados

Fator	Soma dos quadrados
A	104983,61
B	28048,62
AB	27227,22
Total	160259,47

Tabela 11 – Influência de cada fator

Fator	Influências
A	0,65
B	0,18
AB	0,17

Sabendo que a quantidade de navios influencia mais no tempo para responder o BAPTBS pelo Gurobi, foi feita a análise do tempo quando a quantidade de berços se mantêm constante. Aplicando o método dos Mínimos Quadrados Polinomiais no conjunto de dados da Tabela 7 para instâncias com mesmo número de berços, é possível encontrar uma função de grau 5 que representa o comportamento do tempo para este conjunto de instâncias.

As Equações 5.1 a 5.6 são as funções obtidas a partir da regressão polinomial e estão ordenadas em ordem crescente de acordo com o número de berços. O  $x$  nas funções representam os navios e o  $y$  será o valor do tempo para resolver a instância.

$$y = -0.0002508x^5 + 0.0204413x^4 - 0.61065x^3 + 8.53437x^2 - 53.936x + 121.03 \quad (5.1)$$

$$y = -0.0006333921101x^5 + 0.05196407763x^4 - 1.566668423x^3 + 21.71846621x^2 - 136.1957006x + 304.9772425 \quad (5.2)$$

$$y = -0.0002094016217x^5 + 0.0173052955x^4 - 0.5217452837x^3 + 7.244089229x^2 - 45.133708x + 101.7436479 \quad (5.3)$$

$$y = 0.0004828565313x^5 - 0.03550229844x^4 + 0.9949395632x^3 - 13.04680392x^2 + 79.50387522x - 172.6655918 \quad (5.4)$$

$$y = 0.0002937231366x^5 - 0.02243014031x^4 + 0.6673515529x^3 - 9.264977694x^2 + 59.41160612x - 133.1207549 \quad (5.5)$$

$$y = 0.0009007360953x^5 - 0.07259280284x^4 + 2.234693901x^3 - 31.84614503x^2 + 206.6689734x - 471.1297134 \quad (5.6)$$

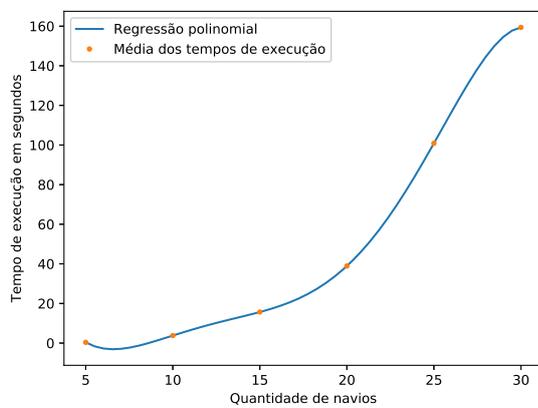


Figura 8 – 1 berço. Fonte: Próprio autor

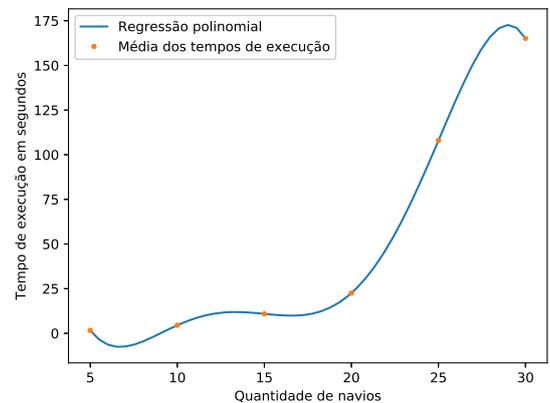


Figura 9 – 2 berços. Fonte: Próprio autor

As Figuras 8 a 13 são os gráficos dos tempos médios de execução quando é variado a quantidade de navios (pontos) e a função obtida pela regressão. É possível observar que a função do tempo em relação ao número de navios tem um crescimento polinomial em todos os casos.

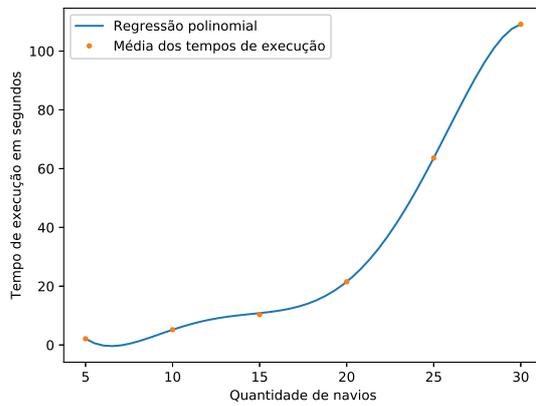


Figura 10 – 3 berços. Fonte: Próprio autor

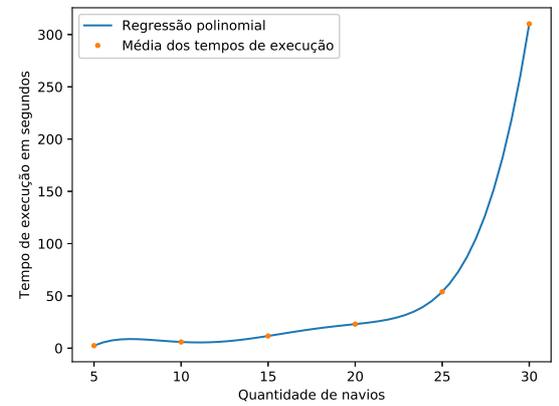


Figura 11 – 4 berços. Fonte: Próprio autor

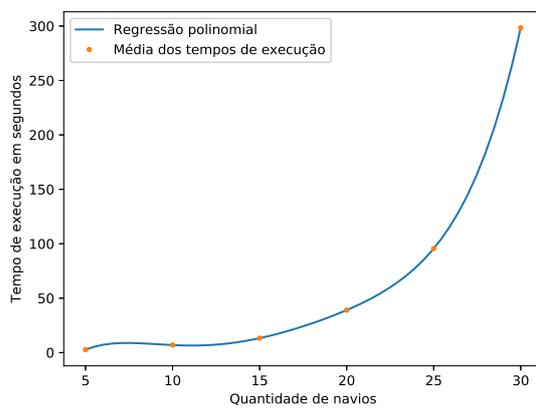


Figura 12 – 5 berços. Fonte: Próprio autor

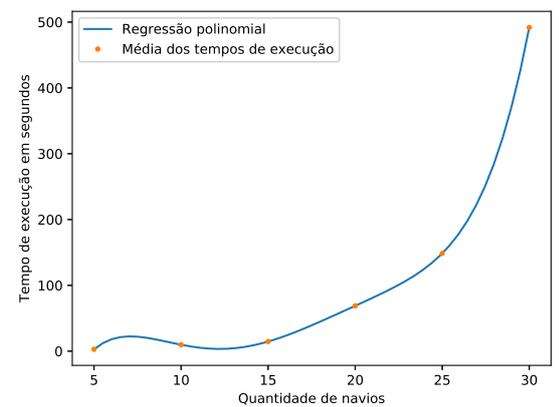


Figura 13 – 6 berços. Fonte: Próprio autor

## 6 Conclusão

O Problema de alocação de berços é amplamente discutido na literatura ([MENEZES, 2016](#)) ([BARROS, 2010](#)) ([BARROS, 2017](#)) ([BARBOSA, 2014](#)) e vários tipos desse problema são encontrados e modelados matematicamente. Este trabalho apresentou o modelo proposto por [Barros \(2017\)](#), onde é proposto o modelo de Problema de Alocação de Berços em Portos Graneleiros com Restrições de Estoque e Condições Favoráveis de Maré.

O Gurobi foi comparado com o solver comercial mais conhecido, o CPLEX, e foi possível notar algumas diferenças no modo de resolução do problema. É possível notar que, na maioria das vezes, quanto melhor for a etapa de presolve, menor será o tempo gasto na tentativa de resolver o *BAPTBS* em tempo viável.

Como contribuição, este trabalho trouxe uma análise de qual atributo, entre quantidade de berços e quantidade de navios, influencia mais para dificultar o problema, logo, aumentando o tempo necessário para que o solve Gurobi consiga resolver o *BAPTBS*. Foi constatado que a quantidade de navios possui 65% de influência na dificuldade do problema. Ao utilizar o Método dos Mínimos Quadrados Polinomias quando a quantidade de berços se mantém constante foi possível obter uma função polinomial de grau 5 para cenários do *BAPTBS* com 1 a 6 berços e é observado que em todos os casos o comportamento do tempo é polinomial.

### 6.1 Trabalhos futuros

Como trabalho futuro, pode-se estudar e implementar técnicas de programação inteira para assim, analisar melhor o que está acontecendo e combinar com heurística para resolver problemas grandes que hoje são inviáveis de serem respondidos apenas com técnicas exatas.

Outras modelagens para o cenário estudado podem ser propostas afim de facilitar o problema. Modelagens de cenários mais completos, que abordem o porto como um todo e não apenas a alocação de berços com condições de maré e estoque também podem ser estudados.

# Referências

- ANDERSEN, E. D.; ANDERSEN, K. D. Presolving in linear programming. *Mathematical Programming*, Springer, v. 71, n. 2, p. 221–245, 1995. Citado na página 8.
- APOLO11.COM. *Tábua de Marés-Previsão de Maré para São Luis / MA*. 2017. Disponível em: <<http://www.apolo11.com/mare.php?local=14>>. Citado na página 5.
- ATRIBUNA. *Porto de Santos responde por 84,8café*. 2017. Disponível em: <<http://www.atribuna.com.br/noticias/noticias-detalle/porto%26mar/porto-de-santos-responde-por-848-das-exportacoes-de-cafe/?cHash=5c7b59f4ad5d9e1ec416d3e339cd5285>>. Citado na página 2.
- BARBOSA, F. *O Problema de Alocação de Berços: Aspectos Teóricos e Computacionais*. Dissertação (Mestrado), 2014. Citado na página 26.
- BARROS, V. H. *PROBLEMA DE ALOCAÇÃO DE BERÇOS EM PORTOS GRANELEIROS COM RESTRIÇÕES DE ESTOQUE E CONDIÇÕES FAVORÁVEIS DE MARÉ*. Dissertação (Mestrado), 2010. Citado 7 vezes nas páginas 11, 3, 4, 5, 16, 22 e 26.
- BARROS, V. H. Estudo do problema de alocação de berços com restrições de maré e controle de estoque com questões de multiobjetividade. 2017. Citado 6 vezes nas páginas 3, 5, 13, 16, 18 e 26.
- BARROS, V. H. et al. Model and heuristic for berth allocation in tidal bulk ports with stock level constraints. *Computers & Industrial Engineering*, Elsevier, v. 60, n. 4, p. 606–613, 2011. Citado na página 3.
- BAZARAA, M. S.; JARVIS, J. J.; SHERALI, H. D. *Linear Programming and network flows*. 4. ed. [S.l.: s.n.], 2010. Citado na página 7.
- BUHRKAL, K. et al. Models for the discrete berth allocation problem: A computational comparison. *Transportation Research Part E: Logistics and Transportation Review*, v. 47, n. 4, p. 461 – 473, 2011. ISSN 1366-5545. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1366554510001201>>. Citado na página 1.
- CORDEAU, J.-F. et al. Models and tabu search heuristics for the berth-allocation problem. *Transportation science*, INFORMS, v. 39, n. 4, p. 526–538, 2005. Citado 2 vezes nas páginas 4 e 5.
- FERGUSON, T. S. *LINEAR PROGRAMMING: A Concise Introduction*. 2000. Disponível em: <<https://www.math.ucla.edu/~tom/LP.pdf>>. Citado na página 7.
- FERNANDES, M. G. *MODELO ECONÔMICO-OPERACIONAL PARA ANÁLISE E DIMENSIONAMENTO DE TERMINAIS DE CONTÊINERES E VEÍCULOS*. Dissertação (Mestrado), 2001. Citado na página 4.
- GUROBI. 2017. Disponível em: <<http://www.gurobi.com>>. Citado 2 vezes nas páginas 3 e 16.

- IBM. 2017. Disponível em: <<http://www.ibm.com>>. Citado na página 16.
- ILOG, Inc. *ILOG CPLEX: High-performance software for mathematical programming and optimization*. 2006. See <<http://www.ilog.com/products/cplex/>>. Citado na página 3.
- ITAQUI. *Movimentação de Carga*. 2017. Disponível em: <<http://www.emap.ma.gov.br/porto-do-itaqui/operacoes-portuarias/movimentacao-de-carga>>. Citado na página 2.
- JAIN, R. K. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. 1. ed. [s.n.], 1991. v. 1. ISBN 0471503363, 9780471503361. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=433a6436756841210c5f20cdf8a7cde1>>. Citado na página 11.
- MENEZES, G. C. *Modelo e algoritmos para um problema integrado de planejamento, sequenciamento e alocação de pátios*. Tese (Doutorado), Setembro 2016. Citado 3 vezes nas páginas 2, 4 e 26.
- RODRIGUES, I. B. G. et al. Modelo matemático para o problema de alocação de berços em portos com limitações de operação de carga ao longo do cais. 2015. Citado 2 vezes nas páginas 11 e 4.
- SANTOS, P. de. *PortoSantos*. 2017. Disponível em: <<http://www.portodesantos.com.br>>. Citado na página 2.
- SCHRIJVER, A. *Theory of linear and integer programming*. 1. ed. [S.l.: s.n.], 1998. Citado na página 8.
- SILVA, A. B. da. *O MÉTODO SIMPLEX E O MÉTODO GRÁFICO NA RESOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO*. Dissertação (Mestrado), 2016. Citado na página 7.
- SIQUEIRA, I. C. dos Santos de. *PrincipaisPortosBR*. 2016. Disponível em: <<https://www.portogente.com.br/portopedia/87168-10-principais-portos-do-brasil-com-infografico>>. Citado na página 1.
- XINHUANET. *Porto de Shanghai mantém 1ª posição em movimentação de contêineres do mundo por sete anos*. 2017. Disponível em: <[http://portuguese.xinhuanet.com/2017-04/26/c\\_136236339\\_3.htm](http://portuguese.xinhuanet.com/2017-04/26/c_136236339_3.htm)>. Citado na página 1.