**Faculdade de Engenharia da Universidade do Porto**



**Sweetland – What Merchant Navy?**

Mestrado Integrado em Engenharia Informática e Computação 2020/2021

Investigação Operacional

**Turma 4MIEIC02 - Grupo 3:**

Ana Marisa Machado Macedo - [up201909572@fe.up.pt](mailto:up201909572@fe.up.pt)

Francisco Costa Filipe de Almeida - [up201909574@fe.up.pt](mailto:up201909574@fe.up.pt)

João Filipe Carvalho de Araújo - up201705577@fe.up.pt

Maria Helena Viegas Oliveira Ferreira - up201704508@fe.up.pt

# Índice

[**Índice**](#_7grvypjqmccz) **2**

[Introdução](#_l4l9nbnh6x4j) **3**

[Itinerários](#_1s6jcivtg5dl) **4**

[Formulação do Modelo](#_12cobd3wu6oy) **6**

[Solução ótima](#_tr6wldty26aa) **11**

[Análise de sensibilidade - Labor cost range](#_in8sv11m11t8) **13**

[Análise de sensibilidade - 15 dias de manutenção por ano](#_7layht9rl37n) **16**

[Formulação modelo - Procura de milho e trigo de Fatland e Sealand duplicam](#_xb0ffe2klu8c) **17**

[Solução ótima](#_ssok95kws81l) **20**

[Relatório de sensibilidade](#_ktscwl8jrcgc) **22**

[Conclusões](#_vh40h4em6p32) **24**

[**Anexos**](#_yxwupik3t409) **25**

## 

# Introdução

O Dr. Peter Hops, Primeiro-Ministro de Sweetland, pretende reestruturar a frota da marinha mercante do seu país, e precisa de **decidir o número de embarcações de diferentes tipos que devem ser adquiridas** e **como as viagens devem ser atribuídas a essas embarcações**, a fim de **satisfazer os requisitos de importação e exportação do país**.

No âmbito da disciplina de Investigação Operacional, no ano letivo de 2020/2021, foi-nos proposto, usando os nossos conhecimentos de modelação, resolução e interpretação de problemas de Programação Linear, ajudar o governo da Sweetland e o seu primeiro-ministro.

Ao longo do relatório o problema é definido mais detalhadamente, é formulado um modelo de programação linear para o resolver e a solução ótima é obtida. São também gerados relatórios de sensibilidade e analisadas certas variáveis e restrições. O processo é repetido, mas considerando que a procura de milho e trigo de Fatland e Sealand duplica.

Para a resolução do problema, implementou-se a modelação em linguagem Python, recorrendo-se à biblioteca **Gurobi**.

# 

# Itinerários

Deve-se, em primeiro lugar, definir os **itinerários** a que os navios podem ser atribuídos, o **tipo de carga** transportada e **tipo de navio**.

Sabe-se *a priori* que, Sweetland tem os portos Doce e Bom, Fatland os portos Sky e Moon e Sealand o porto Mars. Dos portos de Sweetland deve ser exportado **milho** e **trigo**, e importado **ferro** e **cobre** de Fatland e Sealand. Existem dois tipos de navios para o propósito: os pequenos (**Tipo 1**) e os grandes (**Tipo 2**).

Os itinerários devem ter em conta que:

1. Os navios devem partir com as exportações de Doce ou Bom e retornar carregados com as importações de Sky, Moon ou Mars (seja do porto onde descarregaram a carga ou de outro porto em Fatland ou Sealand).
2. As embarcações de grande porte (navio tipo 2) não podem ir para Moon, pois excedem o calado permitido de 8 metros. O calado do navio tipo 1 quando cheio é de 10 metros, enquanto que Moon só permite navios com calado de 8 metros.

Com estas restrições, teríamos 18 itinerários para navios do tipo 1 e 8 itinerários para navios tipo 2. No entanto, sabe-se que Doce só exporta milho e Sky não importa milho, pelo que não pode haver itinerários em que o navio vá diretamente de Doce para Sky. O mesmo acontece com Bom, que só exporta trigo e não importa cobre, e Moon, que não importa trigo e só exporta cobre.

O número de itinerários diminui assim para **12** para o navio tipo 1 e **7** para o navio tipo 2. A tabela seguinte representa os itinerários e devidas cargas:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Itinerário** | **Tipo de navio** | **Produto exportado de Sweetland** | **Produto importado para Sweetland** |
| **1** | Doce – Moon – Doce | 1 | Milho | Cobre |
| **2** | Doce – Moon - Mars - Doce | 1 | Milho | Ferro |
| **3** | Doce – Moon - Sky - Doce | 1 | Milho | Cobre |
| **4** | Doce – Moon - Sky - Doce | 1 | Milho | Ferro |
| **5** | Doce – Mars – Moon – Doce | 1 | Milho | Cobre |
| **6** | Doce – Mars – Doce | 1 e 2 | Milho | Ferro |
| **7** | Doce – Mars – Sky – Doce | 1 e 2 | Milho | Cobre |
| **8** | Doce – Mars – Sky – Doce | 1 e 2 | Milho | Ferro |
| **9** | Bom – Sky – Bom | 1 e 2 | Trigo | Ferro |
| **10** | Bom – Mars – Bom | 1 e 2 | Trigo | Ferro |
| **11** | Bom – Sky – Mars – Bom | 1 e 2 | Trigo | Ferro |
| **12** | Bom – Mars - Sky - Bom | 1 e 2 | Trigo | Ferro |

Tabela 1: Viagens, tipo de navios que podem fazer as respectivas viagens, e produtos exportados e importados por Sweetland.

# Formulação do Modelo

Para formular o problema, devemos definir as **variáveis de decisão**, ou seja, as variáveis sobre as quais vamos decidir os valores, as **restrições**, que impõe limites às decisões, e por fim a **função objetivo**, que define o critério para descobrir a solução ótima.

O modelo encontra-se implementado no ficheiro **“Modelos/P2.py”**. Utiliza-se o método *addVar* para adicionar variáveis de decisão ao modelo, em que um dos parâmetros é o tipo da variável, *vtype*, que pode ser contínua (C) ou inteira (I). Para este modelo usamos variáveis contínuas. Para adicionar restrições ao modelo utiliza-se o método *addConstr, e* para definir a função objetivo do modelo no Gurobi utiliza-se o método *setObjective. Por fim, para encontrar a solução usa-se a função optimize.*

No ficheiro “Modelos/P2.lp” apresenta-se a função objetivo e restrições de uma forma visivelmente mais simples.

**Variáveis de decisão:**

* **vessel1**: número de navios do tipo 1 que devem ser adquiridos;
* **vessel2**: número de navios do tipo 2 que devem ser adquiridos;
* **tripstripType,1**: número de viagens do itinerário tripType (tripType = [1, 12]) que devem ser feitas pelos navios tipo 1. A tripType representa o ID do itinerário apresentado na tabela 1;
* **tripstripType,2**: número de viagens do itinerário tripType (tripType = [6, 12]) que devem ser feitas pelos navios tipo 2. A tripType representa o ID do itinerário apresentado na tabela 1.

De maneira a simplificar a função objetivo, quatro variáveis auxiliares foram criadas.

**Variáveis auxiliares:**

* **dLoadedtype**: distância feita pelos navios do tipo type (type = 1 e 2) enquanto cheios
* **dEmptytype**: distância feita pelos navios do tipo type (type = 1 e 2) enquanto vazios

Este problema é **equilibrado**, ou seja, as necessidades de exportação de Sweetland para trigo e milho são iguais às necessidades de importação de Sealand e Fatland para estas mercadorias, e as necessidades de importação de Sweetland para cobre e ferro são iguais às necessidades de exportação de Sealand e Fatland para estas mercadorias).

Um navio **tipo 1** consegue transportar até **35** **toneladas**, enquanto que um navio **tipo 2** consegue transportar **70 toneladas**.

**Restrições:**

**c1**: 35 \* ( trips9,1 + trips10,1 + trips11,1 + trips12,1)+ 70 \* (trips9,2 + trips10,2 + trips11,2 + trips 12,2) >= 50000

A restrição 1 garante que o número de viagens que **sai de Bom com** **trigo** consegue transportar a quantidade mínima que deve ser exportada de trigo de Bom, i.e., **a capacidade das viagens dos navios têm de ser superior a 50000 toneladas**. Esta restrição é redundante dado o equilíbrio de cargas, especificamente as restrições 7 e 11.

**c2**: 35 \* (trips1,1 + trips2,1 + trips3,1 + trips4,1 + trips5,1 + trips6,1 + trips7,1 + trips8,1) + 70 \* (trips6,2 + trips7,2 + trips8,2) >= 40000

A restrição 2 garante que o número de viagens que **sai de Doce** **com milho** consegue transportar, no mínimo, as **40000** toneladas que Doce deve exportar. Esta restrição é redundante dado o equilíbrio de cargas, especialmente as restrições 8 e a 13.

**c3**: 30 \* (trips9,1 + trips10,1 + trips11,1 + trips12,1) + 70 \* (trips9,2 + trips10,2 + trips11,2 + trips12,2) >= 50000

A restrição 3 garante que o número de viagens que **chegam a Bom com ferro** conseguem transportar as **50000** toneladas que devem ser importadas.

**c4**: 35 \* (trips 1,1 + trips 3,1 + 35 trips 5,1 + 35 trips 7,1) + 70 trips 7,2 >= 20000

A restrição 4 garante que o número de viagens que **importam cobre para Doce** conseguem transportar as **20000** toneladas que devem ser importadas de Doce. Esta restrição é redundante dado as restrições 9 e 12.

**c5**: 35 \* (trips2,1 + trips4,1 + trips6,1 + trips8,1) + 70 (trips6,2 + trips8,2) >= 20000

A restrição 5 garante que o número de viagens que **importam ferro para Doce** conseguem transportar as **20000** toneladas que devem ser importadas.

**c6**: 35 \* (trips2,1 + trips6,1 + trips10,1 + trips11,1) + 70 \* (trips6,2 + trips10,2 + trips11,2) >= 30000

A restrição 6 garante que número de viagens que **retornam a Bom ou Doce** conseguem trazer a quantidade de **ferro** que deve ser **exportada por Mars**, **30000** toneladas.

**c7**: 35 \* (trips10,1 + trips12,1) + 70 \* (trips10,2 + trips12,2) >= 20000

A restrição 7 garante que os navios conseguem levar a quantidade de **trigo** que deve ser **importada por Mars**, **20000** toneladas.

**c8**: 35 \* (trips 5,1 +trips 6,1 + trips 7,1 + trips 8,1 ) + 70 \* ( trips 6,2 + trips 7,2 + trips 8,2 ) >= 10000

A restrição 8 garante que o número de viagens que **levam milho e chegam a Mars** conseguem transportar as **10000** toneladas que devem ser importadas por Mars.

**c9**: 35 \* (trips3,1 + trips7,1) + 70 \* trips7,2 >= 10000

A restrição 9 garante que o número de viagens que **levam cobre e partem de Sky** conseguem transportar as **10000** toneladas que devem ser **exportadas por Sky** e **importadas por Doce**.

**c10**: 35 \* (trips4,1 + trips8,1 + trips9,1 + trips12,1) + 70 \* (trips8,2 + trips9,2 + trips12,2) >= 40000

A restrição 10 garante que o número de viagens que **levam Ferro e partem de Sky** conseguem transportar as **40000** toneladas que devem ser **exportadas**.

**c11**: 35 \* (trips9,1 + trips11,1) + 70 \* (trips9,2 + trips11,2) >= 30000

A restrição 11 garante que o número de viagens que **levam Trigo e chegam a Sky** conseguem transportar as **30000** toneladas que devem ser importadas.

**c12**: 35 \* (trips1,1 + trips5,1) >= 10000

A restrição 12 garante que o número de viagens que **levam cobre e partem de Moon** conseguem transportar as **10000** toneladas que devem ser exportadas.

**c13**: 35 \* (trips1,1 + trips2,1 + trips3,1 + trips4,1) >= 30000

A restrição 13 garante que o número de viagens que **levam milho e chegam a Moon** conseguem transportar as **30000** toneladas que devem ser importadas.

**c14**: TtripType,1\* trips tripType,1 <= vessel1 \* 345 \* 24, para todo o tripType entre 1 e 12

Esta restrição pode ser traduzida nesta:

**c14**: - 8280 vessel1 + 400 trips1,1 + 453.3333333333334 trips2,1 + 456.6666666666666 trips3,1 + 456.6666666666666 trips4,1 + 453.3333333333334 trips5,1 + 440 trips6,1 + 526.6666666666667 trips7,1 + 526.6666666666667 trips8,1 + 480 trips9,1 + 384 trips10,1 + 498.6666666666667 trips11,1 + 498.6666666666667 trips12,1 <= 0

**c15**: TtripType,2\* trips tripType,2 <= vessel2 \* 345 \* 24, para todo o tripType entre 6 e 12

Esta restrição pode ser traduzida nesta:

**c15**: - 8280 vessel2 + 550 trips6,2 + 658.3333333333333 trips7,2 + 658.3333333333333 trips8,2 + 600 trips9,2 + 480 trips10,2 + 623.3333333333333 trips11,2 + 623.3333333333333 trips12,2 <= 0

Na restrição 14 afirmamos que o **tempo total das viagens feitas por navios do tipo 1 é** **inferior ou igual ao número de horas que o navio pode operar por ano multiplicado pelo número de navios necessários do tipo 1**. A restrição 15 é equivalente, mas para os navios do **tipo 2**. Ambas as restrições não estão exatamente corretas matematicamente. Por exemplo, imaginemos o caso em que o tempo operacional do navio é de 30 dias e tem-se 3 vias com a duração de 20 dias. Esta restrições permitiria-nos obter uma solução em que 2 navios iriam fazer as 3 viagens. No entanto, na verdade seriam precisos 3 navios, que usariam apenas 20 dos 30 dias que poderiam estar operacionais. Apesar de não ser completamente precisa matematicamente, num ponto de vista de gestão a restrição faz sentido, e permite-nos reduzir o tempo computacional para chegar a solução.

**c16**: dLoaded1 = 10000 trips1,1 + 10500 trips2,1 + 11000 trips3,1 + 22000 trips4,1 + 10500 trips5,1 + 11000 trips 6,1 + 11500 trips7,1 + 11500 trips8,1 + 12000 trips9,1 + 9600 trips10,1 +10800 trips11,1 +10800 trips12,1

**c17**: dLoaded2 = 11000 trips6,2 + 11500 trips7,2 + 11500 trips8,2 + 12000 trips9,2 + 9600 trips10,2 + 10800 trips 11,2 + 10800 trips 12,2

As restrições 16 e 17 garantem que as variáveis dLoaded1 e dLoaded2 equivalem à **distância percorrida por todos os navios em todas as viagens enquanto estão carregados, para todos os navios tipo 1 e 2 usados**, respectivamente.

**c18**: dEmpty1 = 1000 trips2,1 + 500 trips3,1 + 500 trips4,1 + 1000 trips5,1 + 2000 trips7,1 + 2000 trips8,1 + 2000 trips11,1 + 2000 trips12,1 = 0

**c19**: dEmpty2 = 2000 trips7,2 + 2000 trips8,2 + 2000 trips11,2 + 2000 trips12,2

As restrições 18 e 19 garantem que as variáveis dEmpty1 e dEmpty2 equivalem à **distância percorrida por todos os navios enquanto vazios para todos os navios 1 e 2 usados**, respectivamente.

Em suma, as restrições 1 a 13 garantem que o número de viagens é o correto dado aos valores de importação e exportação fornecidos. Destas restrições, 3 delas são redundantes. As restrições 14 e 15 garantem que o tempo operacional dos navios é de 345 dias. As restrições 16 a 19 garantem a correção do valor das variáveis auxiliares.

**Função Objetivo:**

O custo da **compra dos navios tipo 1** é de 1000 \* 103, enquanto o custo **da compra** **dos navios tipo 2** é de 1500 \* 103. O custo da tripulação (***labor cost***) por ano para os navios **tipo 1** é de 70 \* 103 e do navio **tipo 2** é de 75 \* 103. O **combustível gasto por um navio tipo 1** se estiver **carregado** é de 50 litros/ 1000 km, e de 42 litros/ 1000km se estiver **vazio.** O **combustível gasto por um navio tipo 2** se estiver **carregado** é de 40 litros/ 1000km, se estiver **vazio** reduz para 30. Finalmente, a **compra das unidades é parcelada em prestações fixas sem juros ao longo de 25 anos** e o **custo anual da manutenção é de 10%** do custo inicial de cada navio. Quando definimos que o objetivo é **minimizar o custo no ano inicial**, a função objetivo é a seguinte:

Min 0.1 \* (vessel1 \* 1000000 + vessel2 \* 1500000) + (vessel1 \* 1000000 + vessel2 \* 1500000) / 25 + vessel1 \* 70000 + vessel2 \* 75000 + ((dLoaded1/1000) \* 50 + (dEmpty1/1000) \* 42 + (dLoaded2/1000) \* 40 + (dEmpty2/1000) \* 30) \* 0.8

Se quisermos definir que o objetivo é minimizar os custos ao longo de 25 anos, apenas temos de multiplicar a função objetivo por 25.

# Solução ótima

Com o modelo definido anteriormente, o otimizador do Gurobi encontrou 1 solução, a qual se encontra **“Solution-P2.sol”**.

O número de navios **tipo 1** que são precisos comprar são **46** (45.3), e o número de navios **tipo 2** são **57**.

Os navios **tipo 1** farão **286 viagens** do **itinerário 1** (Doce - Moon - Doce), **286 viagens** do **itinerário 3** (Doce - Moon - Sky - Doce) e **286 viagens** do **itinerário 4** (Doce - Moon - Sky - Doce). Os navios **tipo 2** farão **143 viagens** do **itinerário 6** (Doce - Mars - Doce), **429 viagens** do **itinerário 9** (Bom - Sky - Bom) e **286 viagens** do **itinerário 10** (Doce - Moon - Sky - Doce). Na tabela 10, nos anexos, pode observar-se a diferença de itinerários e nº de viagens entre esta solução e uma solução do mesmo modelo usando variáveis inteiras, em vez de contínuas.

A **distância** percorrida por navios **tipo 1** **carregados** será **12285714.3m** e por **tipo 2** será **9457142.8m**. A distância percorrida por navios **tipo 1** **vazios** será **285714.3m** e por navios **tipo 2** é **0m**.

O custo anual com esta solução é de **26596445.27€**.

Nas tabelas seguintes observam-se as quantidades de exportação e importação dos produtos, em toneladas, e os **fluxos de exportação de milho de Doce** e **de trigo de Bom**, respectivamente**.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Export** | | **Import** | |
| **Wheat** | **Corn** | **Copper** | **Iron** |
| Bom | 50000 | - | - | 50000 |
| Doce | - | 40000 | 20000 | 20000 |

Tabela 2: Quantidades de exportação e importação de Sweetland, em toneladas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Export** | | **Import** | |
| **Copper** | **Iron** | **Wheat** | **Milho** |
| Mars | - | 30000 | 20000 | 10000 |
| Sky | 10000 | 40000 | 30000 | - |
| Moon | 10000 | - | - | 30000 |

Tabela 3: Quantidades de exportação e importação de Sealand e Fatland, em toneladas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID Itinerário** | **Itinerário** | **Navio** | **Espaço (Toneladas)** | **Destino** |
| 1 | Doce - Moon - Doce | 1 | 10 010 | Moon |
| 3 | Doce - Moon - Sky - Doce | 1 | 10 010 | Moon |
| 4 | Doce - Moon - Sky - Doce | 1 | 10 010 | Moon |
| 6 | Doce - Mars - Doce | 2 | 10 010 | Mars |

Tabela 4: Fluxos de exportação de milho de Doce.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID Itinerário** | **Itinerário** | **Navio** | **Espaço (Toneladas)** | **Destino** |
| 9 | Bom - Sky - Bom | 2 | 30 030 | Sky |
| 10 | Bom - Mars - Bom | 2 | 20 020 | Mars |

Tabela 5: Fluxos de exportação de trigo de Bom.

Foi implementado adicionalmente um modelo que restringe o tempo operacional exatamente e utiliza variáveis inteiras. Esse modelo está em **“P2-Exact.py”** e em "P2-Exact.lp" a função objetivo e restrições aplicadas a esse modelo, visivelmente mais simples. O número de navios obtidos foi o mesmo, no entanto as viagens feitas não foram por cada navio. A solução obtida foi próxima, mas o tempo computacional foi muito superior. Portanto, a solução apresentada na pergunta 2 é, portanto, uma boa solução. As viagens feitas acabariam por provocar um custo superior em relação ao obtido com o modelo da pergunta 2, e o tempo necessário para cumprir a restrição temporal exatamente poderia não compensar logisticamente.

# Análise de sensibilidade - *Labor cost range*

O relatório de sensibilidade permite analisar os limites de variação dos vários parâmetros do modelo que permitem manter a solução ótima encontrada, bem como os efeitos provocados na solução pela alteração dos valores desses parâmetros.

Nas figuras seguintes encontra-se o relatório de sensibilidade completo do Modelo definido anteriormente. A geração do relatório encontra-se implementada no ficheiro **“Sensitivity-Reports/Sensitivity-Model2.py”**. O coeficiente de cada variável é obtido através do atributo *Obj,* o *allowable decrease* através do atributo *SAObjLow* e o *allowable increase* através do atributo *SAObjUp.*

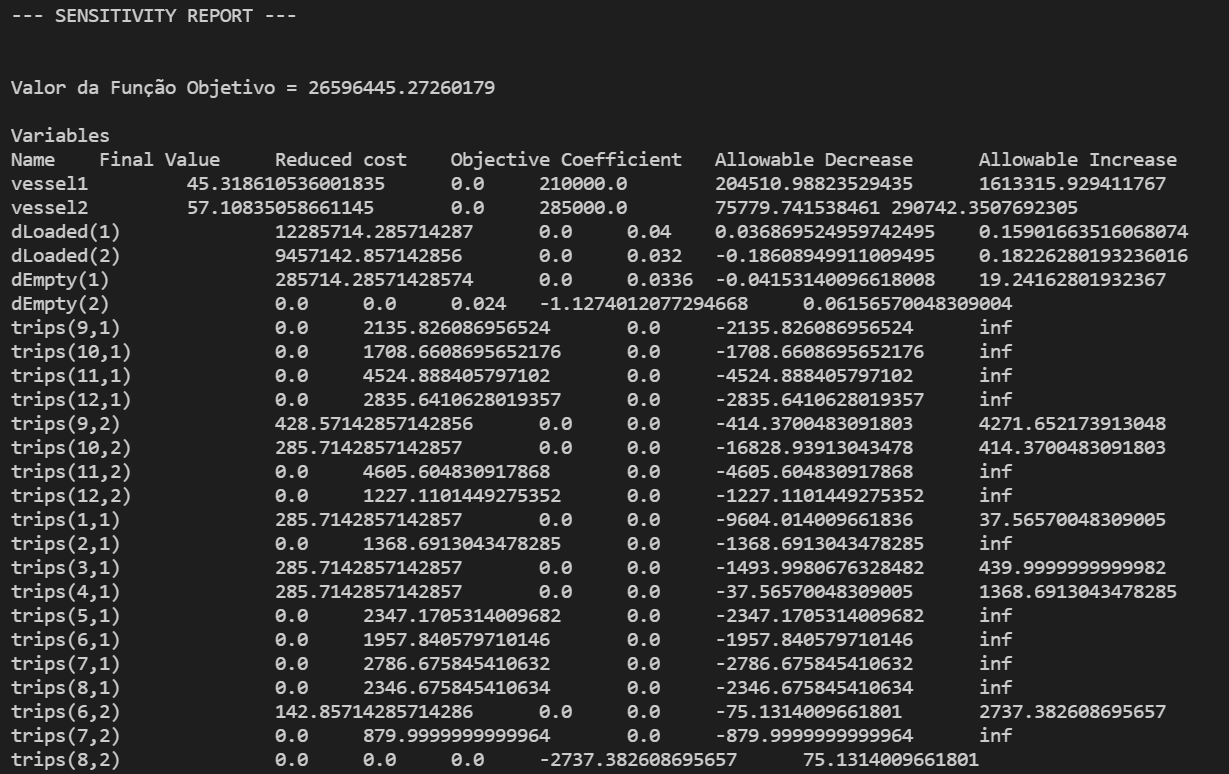


Figura 1: Relatório de sensibilidade das variáveis de decisão do Modelo definido no ficheiro **“Modelos/P2.py”**.

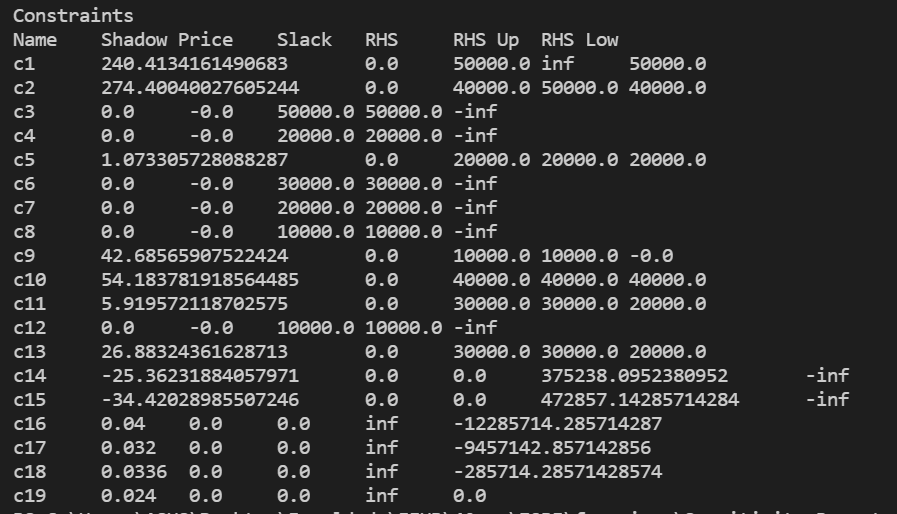


Figura 2: Relatório de sensibilidade das restrições do Modelo definido no ficheiro **“Modelos/P2.py”**.

Através da Figura 1, é possível observar para cada variável de decisão do modelo o *allowable decrease* e *allowable increase* associado ao seu coeficiente. Estes valores representam o intervalo em que o coeficiente da dada variável de decisão na função objectiva pode ser diminuído/aumentado, respetivamente, sem alterar a solução óptima, onde todos os outros dados são fixos.

Por exemplo, o coeficiente da variável **vessel1**, que representa o nº de navios do tipo 1, é de 210000, que equivale a:

0.1 \* 1000000 + 1000000/25 + 70000,

onde 70000 representa o custo da tripulação (***labor cost***). O coeficiente, segundo o relatório de sensibilidade, pode variar entre 204510.988 e 1613315.929, sem que a solução ótima seja alterada. Assim, sendo que o *labor cost* faz parte do coeficiente, para se obter o allowable range desse custo, subtrai-se ou soma-se ao custo total da tripulação (de todos os navios) o allowable decrease ou increase, respectivamente e divide-se o resultado pelo número de navios, ou seja:

*Range Labor cost* inferior = (*labor cost*(total) - *allowable decrease*) / vessel1

= (70 000 \* 46 - 204510.98823529435) / 46 = 65 554,109

Range Labor *cost* superior = (*labor cost* (total) + *allowable increase*) / vessel1

= (70 000 \* 46 + 1613315.929411767) / 46 = 105 072,085

Já para a variável que representa o nº de navios tipo 2, **vessel2**, o seu coeficiente é de 285000.0, que equivale a:

0.1 \* 1500000 + 1500000/25 + 75000,

onde 75000 representa o custo da tripulação (***labor cost***). O coeficiente, segundo o relatório de sensibilidade, pode variar entre 204510.988 e 1613315.929, sem que a solução ótima seja alterada. O cálculo do *range* do *labor cost* para este tipo de navio é equivalente ao cálculo para o navio tipo 1, mas com os novos valores:

*Range Labor cost* inferior = (*labor cost* (total) - *allowable decrease*) / vessel2

= (75 000 \* 57 - 75779.741538461) / 57 = 73 670,531

*Range Labor cost* superior = (*labor cost* (total) + *allowable increase*) / vessel2

= (75 000 \* 57 + 290742.3507692305) / 57 = 80 100,743

Conclui-se então que **o custo da tripulação do navio** **tipo 1** **pode variar** **entre 65554.109 e 105072.085**, sem que o nº ótimo de navios do tipo 1 necessários se altere, e **o custo da tripulação do navio tipo 2 pode variar entre 73670.531 e 80100.743**, sem que o nº ótimo de navios do tipo 2 necessários se altere.

# Análise de sensibilidade - 15 dias de manutenção por ano

O impacto da diminuição da duração de manutenção de navios do tipo 1 no valor total da frota pode ser avaliado através do ***shadow price***da restrição associada ao tempo total de viagem disponível durante um ano para navios do tipo um (restrição c14), visível na figura 2. O valor do *shadow price* é obtido através do atributo *Pi* do objeto que representa a restrição e neste caso corresponde à mudança no custo total da frota por cada hora que varia no tempo total disponível para viagem. O valor do impacto é calculado pela expressão:

shadowPrice \* nDias \* 24h \* nNavios

Neste caso queremos avaliar o impacto quando o número de dias de manutenção de navios do tipo 1 é reduzido de 20 para 15 dias, ou seja, quando há 5 dias adicionais para viagens. O número de navios do tipo 1 é 46, conforme a solução obtida no primeiro modelo. Substituindo os valores na expressão temos:

-25.36231884057971 \* 5 \* 24 \* 46 = - 140 000.00€

Logo, reduzir o tempo de manutenção de navios do tipo 1 em 5 dias resultará na diminuição do custo total da frota em 140 000.00€, que daria 26 456 445.27€.

# Formulação modelo - Procura de milho e trigo de Fatland e Sealand duplicam

Se as exportações totais de Sweetland permanecem inalteradas, mas a procura de milho e trigo de Fatland e Sealand duplicam, de modo a que parte da procura de cereais desses países não pode ser satisfeita por Sweetland, deixamos de ter um problema equilibrado.

Todas as variáveis declaradas na pergunta 2 continuam a existir. Adicionam-se apenas quatro variáveis às existentes, a carga de trigo e milho que chega a Mars, a carga de trigo que chega efetivamente a Sky e a carga de milho que chega a Moon. As restrições de 1 a 6, 9, 10, 12 e de 14 a 19 mantêm-se. Para além dessas, alteram-se as 4 restrições não mantidas (7, 8, 11 e 13) e adicionam-se outras 4. Poderiam acrescentar-se 2 condições não necessárias, que o trigo que chega a Sky é menor que 60000 (que sempre será porque de Bom só chegam 60000 toneladas de trigo) e que o milho que chega a Moon é menor que 60000 (que sempre será porque de Doce só chegam 40000 toneladas de milho).

O novo modelo encontra-se implementado no ficheiro **“Modelos/P6.py”.** Para este modelo usamos variáveis contínuas.

No ficheiro “Modelos/P6.lp” apresenta-se a função objetivo e restrições de uma forma visivelmente mais simples.

**Variáveis de decisão:**

As variáveis vessel1, vessel2, tripstripType,1 , tripstripType,2  descritas anteriormente e as variáveis auxiliaresdLoadedtype , dEmptytype são mantidas. São adicionadas estas 4 novas variáveis:

**load Mars, Trigo:** quantidade de trigo que chega a Mars, i.e., é importada por Mars

**load Mars, Milho**: quantidade de milho que chega a Mars

**load Sky, Trigo**: quantidade de trigo que chega a Sky

**load Moon, Milho**: quantidade de milho que chega a Moon

**Restrições:**

**c7**: 35 (trips10,1 + trips12,1) + 70 (trips10,2 + trips12,2 ) >= loadMars,Trigo

A restrição 7 garante que os navios que fazem as viagens para Mars com Trigo têm espaço para a carga atribuída a Mars, ou seja, a carga que efetivamente chega a Mars.

**c8**: loadMars,Trigo <= 40000

Dado que a procura de Mars por trigo é de 40000 toneladas. A restrição 8 garante que o Trigo que chega a Mars é igual ou inferior a essa quantidade.

**c9**: 35 (trips5,1 + trips6,1 + trips7,1 + trips8,1) + 70 (trips6,2 + trips7,2 + trips8,2 ) >= loadMars,Milho

A restrição 9 garante que os navios que fazem as viagens para Mars com Milho tenham espaço para a carga atribuída a Mars, ou seja, a carga que efetivamente chega a Mars.

**c10**: loadMars,Milho <= 20000

Dado que a procura de Mars por milho duplicou, é agora de 20000 toneladas. A restrição 10 garante que o milho que chega a Mars é igual ou inferior a essa quantidade.

**c13**: 35 (trips9,1 + trips11,1) + 70 (trips9,2 + trips11,2) >= loadSky,Trigo

**c15**: 35 (trips1,1 + trips2,1 + trips3,1 + trips4,1) >= loadMoon,Milho

Dado que a procura de Sky por Trigo e a procura de Moon por milho é superior à oferta, as restrições 13 e 15 são suficientes, garantindo que os navios que fazem as viagens têm espaço para a carga que lhes é atribuída.

**c16**: loadMars,Trigo + loadSky,Trigo = 50000

**c17**: loadMars,Milho + loadMoon,Milho = 40000

Dado que não nos é dito que, se monetariamente compensar, podemos não exportar carga, a carga exportada tem que ser exatamente a carga que pode ser exportada, as 50000 toneladas de Trigo e as 40000 toneladas de milho.

**Restrições descritas anteriormente:**

**c1**: 35 \* ( trips9,1 + trips10,1 + trips11,1 + trips12,1)+ 70 \* (trips9,2 + trips10,2 + trips11,2 + trips 12,2) >= 50000

**c2**: 35 \* (trips1,1 + trips2,1 + trips3,1 + trips4,1 + trips5,1 + trips6,1 + trips7,1 + trips8,1) + 70 \* (trips6,2 + trips7,2 + trips8,2) >= 40000

**c3**: 30 \* (trips9,1 + trips10,1 + trips11,1 + trips12,1) + 70 \* (trips9,2 + trips10,2 + trips11,2 + trips12,2) >= 50000

**c4**: 35 \* (trips 1,1 + trips 3,1 + 35 trips 5,1 + 35 trips 7,1) + 70 trips 7,2 >= 20000

**c5**: 35 \* (trips2,1 + trips4,1 + trips6,1 + trips8,1) + 70 (trips6,2 + trips8,2) >= 20000

**c6:** 35 \* (trips2,1 + trips6,1 + trips10,1 + trips11,1) + 70 \* (trips6,2 + trips10,2 + trips11,2) >= 30000

**c11**: 35 \* (trips3,1 + trips7,1) + 70 \* trips7,2 >= 10000

**c12**: 35 \* (trips4,1 + trips8,1 + trips9,1 + trips12,1) + 70 \* (trips8,2 + trips9,2 + trips12,2) >= 40000

**c14**: 35 \* (trips1,1 + trips5,1) >= 10000

**c18**: TtripType,1\* trips tripType,1 <= vessel1 \* 345 \* 24, para todo o tripType entre 1 e 12

**c19**: TtripType,2\* trips tripType,2 <= vessel2 \* 345 \* 24, para todo o tripType entre 6 e 12

**c20**: dLoaded1 = 10000 trips1,1 + 10500 trips2,1 + 11000 trips3,1 + 22000 trips4,1 + 10500 trips5,1 + 11000 trips 6,1 + 11500 trips7,1 + 11500 trips8,1 + 12000 trips9,1 + 9600 trips10,1 +10800 trips11,1 +10800 trips12,1

**c21**: dLoaded 2 = 11000 trips6,2 + 11500 trips7,2 + 11500 trips8,2 + 12000 trips9,2 + 9600 trips10,2 + 10800 trips 11,2 + 10800 trips 12,2

**c22**: dEmpty1 = 1000 trips2,1 + 500 trips3,1 + 500 trips4,1 + 1000 trips5,1 + 2000 trips7,1 + 2000 trips8,1 + 2000 trips11,1 + 2000 trips12,1 = 0

**c23**: dEmpty2 = 2000 trips7,2 + 2000 trips8,2 + 2000 trips11,2 + 2000 trips12,2

**Função Objetivo:**

Min 0.1 \* (vessel1 \* 1000000 + vessel2 \* 1500000) + (vessel1 \* 1000000 + vessel2 \* 1500000) / 25 + vessel1 \* 70000 + vessel2 \* 75000 + ((dLoaded1/1000) \* 50 + (dEmpty1/1000) \* 42 + (dLoaded2/1000) \* 40 + (dEmpty2/1000) \* 30) \* 0.8

# Solução ótima

Com o modelo definido anteriormente, o otimizador do Gurobi encontrou 1 solução, a qual se encontra **“Solution-P6.sol”**.

O número de navios do **tipo 1** necessários são **30** (29.6) e do **tipo 2** são **69** (68.3). Apesar da frota ser tão diferente, o valor da função objectivo é relativamente próximo. Em vez dos 26375686.4€ obtidos na pergunta 3, teremos **26268417.115€** de gastos. A redução do valor é facilmente explicável, podendo escolher (dentro da procura necessária) que local vamos entregar, iremos realizar as viagens que tiverem menores custos. A carga que chega demonstra esta teoria, pois **a quantidade de trigo que chega a Mars é 30000 toneladas** (em vez de 20000) e **a quantidade de milho que chega a Mars é 20000 toneladas** (em vez de 10000). Portanto, viajar para Mars é mais barato do que viajar para Sky e Moon. Dado que a oferta não mudou, inevitavelmente a oferta que chega a Sky e Moon é menor, sendo que a Sky chegam 20000 toneladas de trigo e a Moon chegam 20000 toneladas de milho. Pode-se confirmar que somando 30000 com 20000 toneladas equivale às 50000 toneladas de trigo que partem de Bom e que somando 20000 com 20000 toneladas equivale às 40000 toneladas de milho que partem de Doce.

Neste cenário, os **navios tipo 1** fazem **286** (285,7) **viagens** do **itinerário 1** e **286** (285,7) do **itinerário 3**. Os navios **tipo 2** fazem agora **286 viagens** do **itinerário 8**, **286 viagens** do **itinerário 9** e **429 viagens** do **itinerário 10**.

Nas tabelas seguintes observam-se as diferenças entre as quantidades de exportação e importação dos produtos, em toneladas, e as diferenças entre os **fluxos de exportação de milho de Doce** e **de trigo de Bom** dos modelos 1 e 2, respectivamente.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Modelo 1** | | | | **Modelo 2** | | | |
|  | **Export** | | **Import** | | **Export** | | **Import** | |
| **Wheat** | **Corn** | **Copper** | **Iron** | **Wheat** | **Corn** | **Copper** | **Iron** |
| Bom | 50000 | - | - | 50000 | 50000 | - | - | 50000 |
| Doce | - | 40000 | 20000 | 20000 | - | 40000 | 20000 | 20000 |

Tabela 6: Diferenças entre as quantidades de exportação e importação de Sweetland, em toneladas.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Modelo 1** | | | | **Modelo 2** | | | |
|  | **Export** | | **Import** | | **Export** | | **Import** | |
| **Copper** | **Iron** | **Wheat** | **Milho** | **Copper** | **Iron** | **Wheat** | **Milho** |
| Mars | - | 30000 | **20000** | **10000** | - | 30000 | **30000** | **20000** |
| Sky | 10000 | 40000 | **30000** | - | 10000 | 40000 | **20000** | - |
| Moon | 10000 | - | - | **30000** | 10000 | - | - | **20000** |

Tabela 7: Diferenças entre as quantidades de exportação e importação de Sealand e Fatland, em toneladas.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Modelo 1** | | | | | **Modelo 2** | | | | |
| **ID Itinerário** | **Tipo de navio** | **Espaço (Tons)** | **Carga**  **(Tons)** | **Destino** | **ID Itinerário** | **Tipo de navio** | **Espaço (Tons)** | **Carga**  **(Tons)** | **Destino** |
| 1 | 1 | 10 010 | 10 000 | Moon | 1 | 1 | 10 010 | 10 000 | Moon |
| 3 | 1 | 10 010 | 10 000 | Moon | 3 | 1 | 10 010 | 10 000 | Moon |
| 4 | 1 | 10 010 | 10 000 | Moon | - | - | - | - | - |
| 6 | 2 | 10 010 | 10 000 | Mars | - | - | - | - | - |
| - | - | - |  | - | 8 | 2 | 20 020 | 20 000 | Mars |

Tabela 8: Diferenças entre os fluxos de exportação de milho de Doce dos modelos 1 e 2, em termos de viagens.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Modelo 1** | | | | | **Modelo 2** | | | | |
| **ID Itinerário** | **Navio** | **Espaço (Tons)** | **Carga**  **(Tons)** | **Destino** | **ID Itinerário** | **Navio** | **Espaço (Tons)** | **Carga**  **(Tons)** | **Destino** |
| 9 | 2 | 30 030 | 30 000 | Sky | 9 | 2 | 20 020 | 20 000 | Sky |
| 10 | 2 | 20 020 | 20 000 | Mars | 10 | 2 | 30 030 | 30 000 | Mars |

Tabela 9: Diferenças entre os fluxos de exportação de trigo de Bom dos modelos 1 e 2.

Nos anexos, na tabela 11, apresenta-se ainda a diferença entre o nº de viagens feitas em cada modelo para cada itinerário.

# Relatório de sensibilidade

No ficheiro **“Sensitivity-Reports\Sensitivity-Model7.py”** está implementada a geração do relatório de sensibilidade do Modelo definido no capítulo anterior. Nas figuras seguintes pode observar-se o resultado.

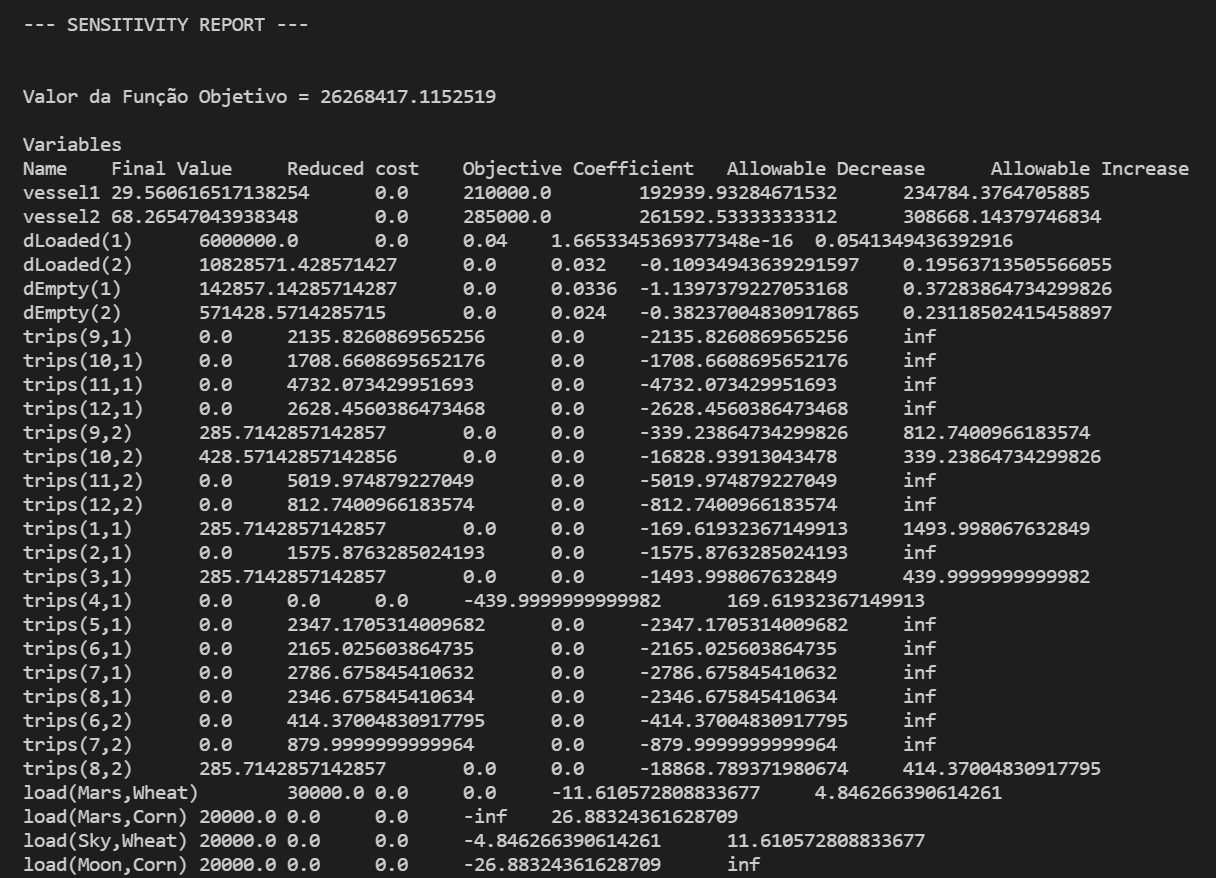


Figura 3: Relatório de sensibilidade das variáveis do Modelo definido no ficheiro **“Modelos/P6.py”**.

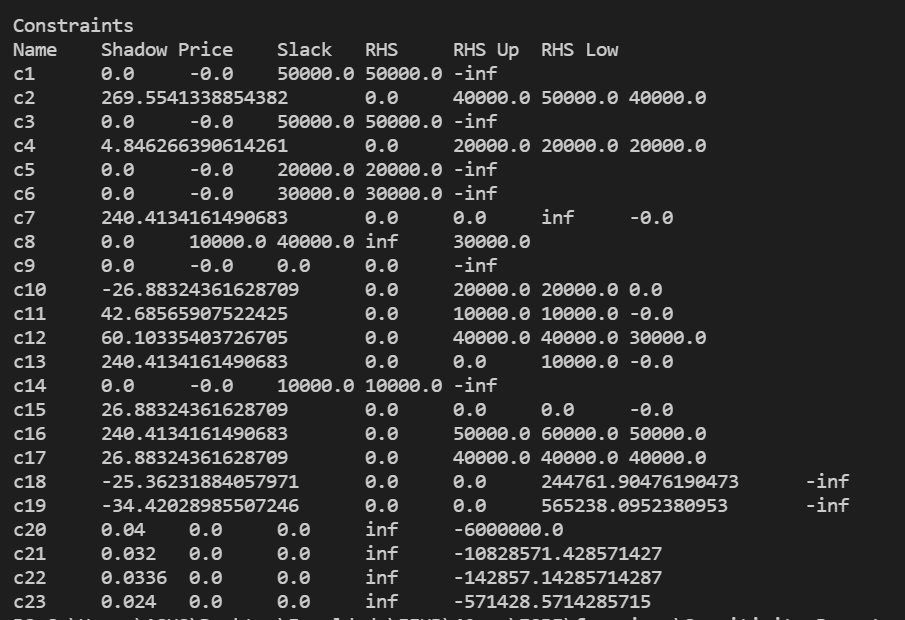


Figura 4: Relatório de sensibilidade das restrições do Modelo definido no ficheiro **“Modelos/P6.py”**.

Da pergunta 6 sabemos que a quantidade de milho que chega a Mars são as 20000 toneladas desejadas. No contexto do modelo proposto na Pergunta 6, o efeito provocado no custo total da frota pelo aumento da quantidade de milho importado por Mars pode ser avaliado pelo *shadow price* da restrição associada a esta quantidade (c10). Neste caso queremos avaliar o impacto quando o valor de milho importado por Mars passa de 20000 para 25000 toneladas, isto é, quando a procura aumenta em 5000 toneladas. Tal consegue-se através da expressão:

shadowPrice \* 5000

Substituindo o valor do *shadow price* tem-se:

-26.883243616287093 \* 5000 = -134416.2181€

Logo, se a quantidade de milho importado por Mars passar para 25000 o custo total da frota diminui em 134416.2181€, resultando em 26134000.8969€.

# Conclusões

Neste trabalho tivemos a oportunidade de aplicar os conhecimentos adquiridos nas aulas de investigação operacional, contribuindo assim para aprofundar e consolidar melhor os conhecimentos obtidos com a disciplina. O trabalho correu como esperado e os resultados obtidos foram os desejados.

# Anexos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Primeiro modelo** | | | |
|  | **Variáveis contínuas** | | **Variáveis inteiras** | |
| ID Itinerários | navios tipo 1 (46) | navios tipo 2 (57) | navios tipo 1 (46) | navios tipo 2 (57) |
| 1 | 286 | na | 286 | na |
| 2 | **0** | na | **37** | na |
| 3 | 286 | na | 286 | na |
| 4 | **286** | na | **249** | na |
| 5 | 0 | na | 0 | na |
| 6 | **0** | **143** | **10** | **120** |
| 7 | 0 | 0 | 0 | 0 |
| 8 | 0 | **0** | 0 | **18** |
| 9 | **0** | **429** | **2** | **428** |
| 10 | 0 | 286 | 0 | 286 |
| 11 e 12 | 0 | 0 | 0 | 0 |

Tabela 10: Diferença de itinerários feitos e nº de viagens de cada itinerário entre as soluções do primeiro modelo usando variáveis contínuas ou inteiras

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Primeiro modelo** | | **Segundo modelo** | |
| ID Itinerários | navios tipo 1 (46) | navios tipo 2 (57) | navios tipo 1 (33) | navios tipo 2 (66) |
| 1 | 286 | na | 286 | na |
| 2 | 0 | na | 0 | na |
| 3 | 286 | na | 286 | na |
| 4 | **286** | na | **0** | na |
| 5 | 0 | na | 0 | na |
| 6 | 0 | **143** | 0 | **0** |
| 7 | 0 | 0 | 0 | 0 |
| 8 | 0 | **0** | 0 | **286** |
| 9 | 0 | **429** | 0 | **286** |
| 10 | 0 | **286** | 0 | **429** |
| 11 e 12 | 0 | 0 | 0 | 0 |

Tabela 11: Diferença de itinerários feitos e nº de viagens de cada itinerário entre as soluções do primeiro e segundo modelo