

Universidade do Minho  
Escola de Engenharia  
Mestrado Integrado em Engenharia de Telecomunicações e Informática  
Investigação Operacional

# **O Armazém de Paletes**

Bruno Xavier Brás dos Santos A72122

Mário Pedro Magalhães Marinho A85957

Pedro Alexandre Gomes da Silva Pereira A83659

Guimarães  
10 de Junho de 2019

## **Resumo**

“Productivity growth is the only possible way to achieve prosperity, to create a solid sustainable foundation”<sup>1</sup>

O presente relatório divide-se em vários capítulos, cada capítulo está repartido em secções, e por cada secção existe algumas subsecções. A organização dos capítulos está baseada da seguinte forma:

No primeiro capítulo colocamos a introdução.

No segundo capítulo colocamos uma descrição informal do projeto, os dados mais relevantes e os requisitos requeridos.

O terceiro capítulo descreve de forma sucinta a fase 1 do projeto. Contem uma pequena descrição do problema, o ficheiro '.mod' usado e os resultados.

No quarto capítulo, colocamos os dados da fase 2.

No quinto capítulo foi dedicado à fase 3.

No sexto capítulo colocamos a conclusão.

No último capítulo colocamos as referências bibliográficas dos conteúdos que nos ajudaram a realizar este projeto.

A última parte do relatório são os apêndices.

---

<sup>1</sup><https://www.thomaswhite.com/global-perspectives/mario-draghi-president-designate-european-central-bank/>

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Análise e Especificação</b>	<b>4</b>
2.1	Contextualização do problema . . . . .	4
2.2	Especificação dos Requisitos . . . . .	5
2.2.1	Constituição do armazém . . . . .	5
2.2.2	Dimensões da base da paleta - estrado de madeira . . . . .	6
2.2.3	Dados . . . . .	6
<b>3</b>	<b>Fase I - Problema de Afetação</b>	<b>8</b>
3.1	Descrição do problema . . . . .	8
3.2	ficheiro '.mod' usado na fase 1 . . . . .	8
3.3	Desenho da resolução . . . . .	9
3.4	Resultados - instância de 10 paletes . . . . .	9
3.5	Resultados - instância de 50 paletes . . . . .	10
3.6	Resultados - instância de 500 paletes . . . . .	11
<b>4</b>	<b>Fase II - (Variante) Problema de Afetação Generalizado (Binário)</b>	<b>13</b>
4.1	Descrição do problema . . . . .	13
4.2	ficheiro '.mod' usado na fase 2 . . . . .	13
4.3	Desenho da resolução . . . . .	14
4.4	Resultados - instância de 10 paletes . . . . .	14
4.5	Resultados - instância de 50 paletes . . . . .	16
4.6	Resultados - instância de 500 paletes . . . . .	17
<b>5</b>	<b>Fase III - Problema de Afetação e Layout de Armazém</b>	<b>19</b>
5.1	Descrição do problema . . . . .	19
5.2	ficheiro '.mod' usado na fase 3 . . . . .	19
5.3	Desenho da resolução . . . . .	21
5.4	Resultados - instância de 10 paletes . . . . .	21
5.5	Resultados - instância de 50 paletes . . . . .	24
5.6	Resultados - instância de 500 paletes . . . . .	28
<b>6</b>	<b>Conclusão</b>	<b>31</b>
<b>7</b>	<b>Bibliografia</b>	<b>32</b>
<b>A</b>	<b>Layouts dos armazéns</b>	<b>33</b>
A.1	Instância de 10 paletes . . . . .	34
A.2	Instância de 50 paletes . . . . .	35

A.3	Instância de 500 paletes . . . . .	36
<b>B</b>	<b>Outros anexos</b>	<b>37</b>

# Capítulo 1

## Introdução

O presente relatório foi realizado para um projeto da unidade curricular de investigação operacional do 2º ano do curso de mestrado integrado em Engenharia de Telecomunicações e Informática da Universidade do Minho, sendo este o terceiro método de avaliação.

Os nossos objetivos para este projeto são: aplicar os conceitos abordados na sala de aula, adquirir metodologias de trabalho individuais e em equipa, e elaborar adquirir ferramentas e competências na temática da investigação operacional.

O projeto consiste na guarda de paletes num armazém. O projeto é dividido em três fases, por ordem crescente de complexidade. Todas as questões retratados no enunciado encontram-se em detalhado nas secções "Descrição do problema" em cada fase do projeto.

Para o projeto, tivemos que implementar vários layouts de armazéns e fazer testes. Na fase três, os nossos armazéns adquiriram uma componente dinâmica, no sentido em que cada nível podia alterar de posição.

Todos os resultados obtidos e mencionados no relatório, encontram-se na pasta 'anexos' que acompanhou o presente relatório.

Para a elaboração do presente relatório, optamos por utilizar o software  $\text{\LaTeX}$ . Sendo este software uma excelente forma de produzir trabalhos científicos, relatórios e textos académicos com excelente qualidade..

## Capítulo 2

# Análise e Especificação

### 2.1 Contextualização do problema

Para a resolução do presente projeto, provimos de um enunciado, no qual estava abordado, detalhadamente, as diferentes fases e os requisitos requeridos das respectivas fases.

Este trabalho prático tem como objetivo a simulação de um armazém, mais concretamente, a otimização dos espaços disponíveis num armazém.

O armazém considerado é dedicado à guarda de paletes. Uma paleta é caracterizada por um conjunto de características: largura  $X$ , profundidade  $Y$ , altura  $Z$  e utilização anual  $W$ . A Figura 1 ilustra a forma típica de uma paleta considerada neste projeto.

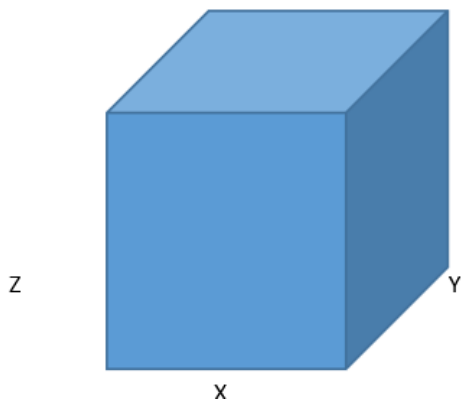


Figura 2.1: Dimensões de uma paleta.

A paleta (o seu conteúdo) é colocada numa posição de um nível (ou alvéolo) de uma RACK que tem uma dada largura  $L$ , profundidade  $Y$  (simples - idêntica à das paletes), e uma altura  $A$  (configurável na Fase II e na Fase III).

As paletes, realisticamente, têm restrições. No nosso projeto, as restrições consideradas desde o início são:

1.  $3X$  menor ou igual a  $L$  (no caso de paleta industrial)
2.  $4X$  menor ou igual a  $L$  (no caso de europalete)
3.  $Z$  menor ou igual a  $A$

Ao longo da elaboração do projeto, consideramos diferentes configurações do armazém, estas diferentes configurações resultam em diferentes problemas de otimização, mas sempre com capacidade suficiente para armazenar a totalidade de paletes existentes em cada situação. Iremos estudar diferentes dimensões para o número de paletes: 10, 50 e 500 paletes.

A Unidade de Medida (UM) a considerar no projeto é o 'metro linear que se mede em função da medida  $X$  das paletes que é necessário armazenar, e da medida  $A$  (altura) dos níveis das RACKS.

Para o tempo, usamos o segundo, mas sempre que achamos necessário, convertemos para horas ou dias.

Em relação à altura máxima de cada rack, e depois de algumas pesquisas, definimos como 600cm, ou seja, 6 metros.

Em relação aos tempos de acesso a cada nível, e visto que optamos por um armazém em que a colocação das paletes é feita de forma automática, e que a máquina responsável por tal demora 1 segundo em ir para o próximo nível.

Este projeto é constituído por 3 fases, de ordem crescente de complexidade, nas quais vamos simular a resposta do armazém para os problemas requeridos das respetivas fases.

## 2.2 Especificação dos Requisitos

### 2.2.1 Constituição do armazém

O armazém mais simples que podemos construir é constituído por uma porta e um rack que contem apenas um nível.

Para os diferentes layouts e fases do nosso projeto vamos considerar que o nosso armazém contém apenas uma porta.

Na figura 2.2 existem dois racks. Cada rack é constituído por 4 níveis, se considerarmos a parte superior do rack um nível. É em cada nível onde as paletes serão armazenadas. Para os diferentes layouts do nosso projeto, o número de racks e de níveis em cada rack serão diferentes.

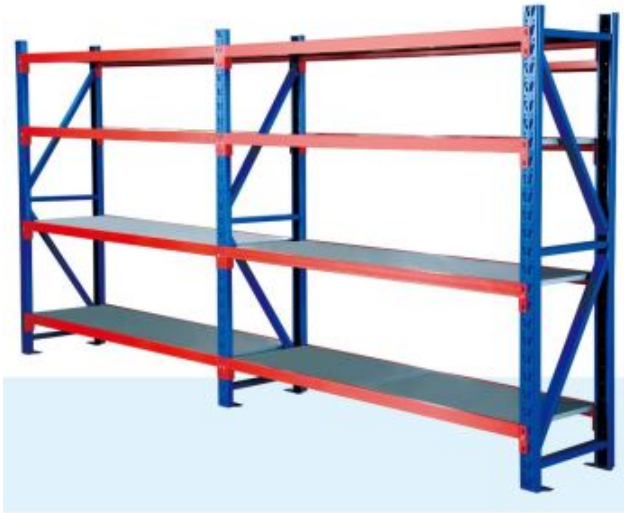


Figura 2.2: exemplo de dois pequenos racks.

Um armazém contém uma porta e corredores que separam alinhamentos de racks. Cada alinhamento pode ser constituído por vários racks.

Na figura 2.2 temos a vista superior de um armazém, com 6 alinhamentos e três corredores. Cada alinhamento contém 15 racks.

Alinhamento 1	RACK_1	RACK_2	RACK_3	RACK_4	RACK_5	RACK_6	RACK_7	RACK_8	RACK_9	RACK_10	RACK_11	RACK_12	RACK_13	RACK_14	RACK_15
PORTA	1														
Alinhamento 2	RACK_30	RACK_29	RACK_28	RACK_27	RACK_26	RACK_25	RACK_24	RACK_23	RACK_22	RACK_21	RACK_20	RACK_19	RACK_18	RACK_17	RACK_16
Alinhamento 3	RACK_31	RACK_32	RACK_33	RACK_34	RACK_35	RACK_36	RACK_37	RACK_38	RACK_39	RACK_40	RACK_41	RACK_42	RACK_43	RACK_44	RACK_45
4															
Alinhamento 4	RACK_60	RACK_59	RACK_58	RACK_57	RACK_56	RACK_55	RACK_54	RACK_53	RACK_52	RACK_51	RACK_50	RACK_49	RACK_48	RACK_47	RACK_46
Alinhamento 5	RACK_61	RACK_62	RACK_63	RACK_64	RACK_65	RACK_66	RACK_67	RACK_68	RACK_69	RACK_70	RACK_71	RACK_72	RACK_73	RACK_74	RACK_75
8															
Alinhamento 6	RACK_90	RACK_89	RACK_88	RACK_87	RACK_86	RACK_85	RACK_84	RACK_83	RACK_82	RACK_81	RACK_80	RACK_79	RACK_78	RACK_77	RACK_76

Figura 2.3: Vista superior de um conjunto de 3 corredores com 6 alinhamentos de RACKs.

## 2.2.2 Dimensões da base da paleta - estrado de madeira

Em termos de estrado / base de madeira para constituição de uma paleta com os produtos do cliente há várias medidas disponíveis. Neste projeto iremos considerar dois tipos de paleta que diferem essencialmente na medida X (largura da paleta): paleta industrial (ou americana) e paleta europeia (ou europaleta).

As medidas consideradas neste projeto, para os dois tipos de paletes, são:

Tabela 2.1: medidas das paletes

Medidas	Paleta Industrial	Europaleta
Dimensões exteriores largura	1000 mm	800 mm
Dimensões exteriores profundidade	1200 mm	1200 mm
Dimensões exteriores altura	144 mm	144 mm
Capacidade de carga dinâmica	1000 kg	1500 kg



Figura 2.4: Paleta Industrial



Figura 2.5: Europaleta

Neste projeto, provavelmente, não será estudado o efeito do peso das paletes.

## 2.2.3 Dados

O cerne deste projeto corresponde à otimização na distribuição de um conjunto de paletes pelos racks, que contem diversos níveis, de cada alinhamento do layout do armazém projetado pelo nosso grupo.

Existem dois tipos de paletes, a paleta industrial e a europaleta.

Uma paleta é identificada por 4 campos: referência, altura, largura e taxa de utilização.

A referência corresponde ao seu número de identificação. Cada paleta tem uma referência que distingue das outras. Uma referência é única e identifica a paleta em questão.

A altura, é a altura do estrado de madeira somado com a altura da mercadoria embalada.

A largura é a largura da paleta. A largura da mercadoria não deve exceder a largura do estrado da madeira da paleta. Para a europaleta, a sua largura é 800 mm. No caso da paleta industrial, a sua largura é 1000 mm.

A taxa de utilização corresponde ao número de movimentos que é feito pela paleta com a respetiva referência. Por exemplo, se a referência 258 possuir uma taxa de utilização de 29 e a referência 301 uma taxa de utilização 45, concluímos que a paleta cuja referência 258 é mais movimentada que a paleta com referência 301.

Cada nível possui dois campos com interesse no nosso projeto. A acessibilidade e a largura.

A acessibilidade corresponde ao tempo gasto desde a porta do armazém até ao respetivo nível. Cada nível é numerado de 1 até à quantidade de níveis existentes no armazém.

A largura, corresponde à largura do respetivo nível. Uma paleta inserida num nível, terá que, obrigatoriamente, ter uma largura inferior ou igual à largura do nível em que irá ser colocada.



Resumo:

Palete: referencia, altura, largura, taxa de utilização;

Nível: acessibilidade, largura;

## Capítulo 3

# Fase I - Problema de Afetação

### 3.1 Descrição do problema

Neste problema pretendia-se que fosse minimizado a área de armazenamento disponível no armazém para a recolha de paletes, considerando que cada paleta é armazenado numa posição do armazém, e cada posição só pode armazenar uma paleta. A Figura 3.1 ilustra o armazenamento no alvéolo de uma única paleta.

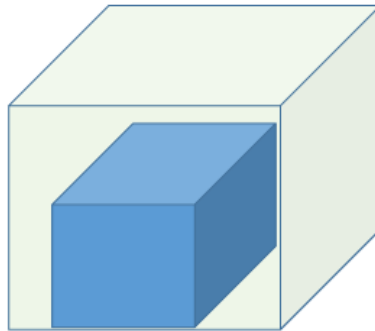


Figura 3.1: RACKs só com uma paleta por nível / alvéolo.

O objetivo da fase 1 é modelar e obter uma solução que faça a afetação das paletes às posições existentes nos níveis (alvéolos) das RACKs do armazém de forma a minimizar a área de armazenamento disponível.

Nesta fase, foi ignorada a altura das paletes, assumindo que a altura do nível na RACK é suficiente para armazenar qualquer paleta, isto é, é igual ou maior que a altura da paleta mais alta. A profundidade da RACK coincide com medida Y das paletes. Os parâmetros importantes das paletes utilizados nesta fase foram a sua taxa de utilização e a sua referencia. Nesta primeira fase, a largura de cada nível foi constante e igual a 1000 cm, ou seja, cada nível só podia ser preenchido por uma e uma só paleta.

### 3.2 ficheiro '.mod' usado na fase 1

Na figura 3.2 encontra-se o ficheiro .mod usado na fase 1.

A linha (1) e (3) definem os parâmetros que são lidos do ficheiro '.dat'.

As linhas (5) e (6) definem que P e N correspondem, respetivamente, às paletes e níveis.

Na linha (11) definimos uma variável,  $x[i,j]$ , binária, que corresponde à colocação da paleta  $i$  no nível  $j$ .

Na linha (13) encontra-se a função objetivo, denominada 'TempoTotal'. O objetivo é minimizar o tempo total despendido na atribuição de cada paleta a cada nível.

Nas linhas (15) e (16) correspondem às restrições que são respeitadas. Na linha (15), 'AfetarPaleta', indica que cada paleta tem que ser atribuída a um nível. na linha (16), a restrição 'OcuparNivel',

indica que cada nível só pode conter uma e uma só paleta.

```
1 param N_paletes;  
2  
3 param N_niveis;  
4  
5 set P := {i in 1..N_paletes};  
6 set N := {j in 1..N_niveis};  
7  
8 param Paleta {i in P, j in 1..4};  
9 param Nivel {i in N, j in 1..2};  
10  
11 var x {i in P, j in N}, binary;  
12  
13 minimize TempoTotal : sum {i in P, j in N} Paleta[i,4] * Nivel[j,1] * x[i,j];  
14  
15 s.t. AfetarPaleta {i in P} : sum {j in N} x[i,j]=1;  
16 s.t. OcuparNivel {j in N} : sum {i in P} x[i,j]<=1;
```

Figura 3.2: ficheiro '.mod' usado na fase 1

### 3.3 Desenho da resolução

O objetivo do problema é modelar e obter uma solução que faça a afetação das paletes às posições existentes nos níveis (alvéolos). Nesta fase, simulamos três instâncias diferentes (10, 50 e 100) de paletes. Para cada instância de paletes, desenhamos três layouts diferentes de armazéns. Na fase 1,

Para a fase 1, desenhamos três layouts de armazéns diferentes para cada uma das três instâncias de paletes (10 paletes, 50 paletes e 500 paletes). No anexo A, bem como em formato digital, encontra-se os nove layouts usados, dos quais os layouts 1,2 e 3 são usados para simular a instância de 10 paletes, os layouts 4,5 e 6 são usados para simular a instância de 50 paletes e os layouts 7,8 e 9 são usados para simular a instância de 500 paletes.

### 3.4 Resultados - instância de 10 paletes

Para a instância de 10 paletes, foram concebidos três layouts diferentes de armazéns e comparamos os resultados obtidos para estes três layouts.

A instância de 10 paletes usada na simulação foi a mesma para os três layouts e pode-se encontrar em anexos.

No anexo A, nas figuras A.1, A.2 e A.3 encontra-se os layouts 1, 2 e 3, respetivamente.

Na pasta 'anexos', mais concretamente no 'path': 'anexos/resultados/fase 1/instancia 10 paletes', encontra-se os resultados de cada uma das três simulações dos três layouts para a mesma instância de 10 paletes.

Para o layout 1, o valor da função objetivo foi de 813. O que podemos concluir que, para o layout 1, são precisos 813 segundos para a afetação das 10 paletes estar concluída.

No caso do layout 2, o valor da função objetivo é de 686. Para o layout 3, o seu valor é de 731.

Com base nos resultados, concluímos que o layout 2 do armazém é um layout que produz um menor valor de tempo na alocação das paletes em cada nível. Na figura 3.3 é apresentado um pequeno excerto dos resultados da simulação do layout 2, no qual podemos ver que obtivemos uma solução ótima, cujo valor é de 686.

```

100 variables, all binary
20 constraints, all linear; 200 nonzeros
    10 equality constraints
    10 inequality constraints
1 linear objective; 100 nonzeros.

Gurobi 8.1.0: threads=4
Gurobi 8.1.0: optimal solution; objective 686
40 simplex iterations
:   _varname[j] _var[j]   :=
1   'x[1,1]'      1
..   ..

```

Figura 3.3: Resultado layout 2 instância de 10 paletes

### 3.5 Resultados - instância de 50 paletes

Para a instância de 50 paletes, foram concebidos três layouts diferentes de armazéns e comparamos os resultados obtidos para estes três layouts.

A instância de 50 paletes usada na simulação foi a mesma para os três layouts e pode-se encontrar em anexos.

No anexo A, nas figuras A.4, A.5 e A.6 encontra-se os layouts 4, 5 e 6, respetivamente.

Para o layout 4, o valor da função objetivo foi de 2977. O que podemos concluir que, para o layout 1, são precisos 2977 segundos para a afetação das 10 paletes estar concluída.

No caso do layout 5, o valor da função objetivo é de 1634. Para o layout 6, o seu valor é de 2029.

Analisando o resultado, e se continuar com o mesmo critério de avaliação de um layout - menor tempo na afetação das paletes - podemos concluir que o layout 5 é o que apresenta um menor valor, 1634, o que indica, e admitindo que os parâmetros indicados no início são os mesmos, para colocar todas as paletes em níveis separados, são necessários 1634 segundos. O layout 4, neste exemplo, mostrou ser um layout descartado pelos layouts 5 e 6.

Ao analisar os layouts, antes de analisar os resultados concretamente, é intuitivo que o layout 5 irá ter melhores resultados, uma vez que é o que possui mais alinhamentos e um maior número de níveis em cada rack, resultando num menor tempo de acesso a cada nível no geral.

A figura 3.4 mostra um pequeno excerto dos resultados do layout 5. Podemos ver que obtivemos uma solução ótima, de valor 1634. Existem 2656 variáveis  $x[i,j]$ . Também podemos ver em qual nível deve ser colocada uma paleta. Por exemplo, a variável número 10 indica que a paleta 1 deve ser colocada no nível 10, a paleta 2 no nível 1.

```

2646 variables, all binary
103 constraints, all linear; 5292 nonzeros
    49 equality constraints
    54 inequality constraints
1 linear objective; 2646 nonzeros.

Gurobi 8.1.0: threads=4
Gurobi 8.1.0: optimal solution; objective 1634
396 simplex iterations
:      _varname[j] _var[j]      :=
10      'x[1,10]'      1
55      'x[2,1]'      1
110     'x[3,2]'      1
181     'x[4,19]'     1
244     'x[5,28]'     1
274     'x[6,4]'      1
337     'x[7,13]'     1
389     'x[8,11]'     1
463     'x[9,31]'     1
508     'x[10,22]'    1
554     'x[11,14]'    1
610     'x[12,16]'    1
655     'x[13,7]'     1
748     'x[14,46]'    1
759     'x[15,3]'     1
839     'x[16,29]'    1
901     'x[17,37]'    1
923     'x[18,5]'     1
995     'x[19,23]'    1
1060    'x[20,34]'    1
1100    'x[21,20]'    1

```

Figura 3.4: Resultado layout 5 instância de 50 paletes

### 3.6 Resultados - instância de 500 paletes

Do mesmo modo que nos anteriores resultados, para a instancia de 500 paletes foram concebidos três layouts diferentes de armazéns e comparados os resultados obtidos para estes três layouts.

A instância de 500 paletes usada na simulação foi a mesma para os três layouts e pode-se encontrar em anexos.

No anexo A, na figura A.7 encontra-se o layout 8. Os layouts 7 e 9, devido às suas características, só foi possível colocar em formato digital na pasta anexos.

Para o layout 7, o valor da função objetivo foi de 422941. O que podemos concluir que, para o layout 7, são precisos 422941 segundos para a afetação das 500 paletes estar concluída.

No caso do layout 8, o valor da função objetivo é de 97309. Para o layout 9, o seu valor é de 213199.

Analisando o resultado, e se continuar com o mesmo critério de avaliação de um layout - menor tempo na afetação das paletes - podemos concluir que o layout 8 é o que apresenta um menor valor, 97309, o que indica, e admitindo que os parâmetros indicados no início são os mesmos, para colocar todas as paletes em níveis separados, são necessários 97309 segundos.

Nesta simulação, o valor da função objetivo do layout 7 foi de 422941. Para o layout 7, para a afetação de todas as paletes, são necessários 422941 segundos, ou seja, são necessários, aproximadamente 7 059 minutos, ou 118 horas. Para um dia de trabalho de oito horas, e para uma única máquina, são necessários, aproximadamente, 15 dias para uma afetação completa das 500 paletes. Em

```

226000 variables, all binary
952 constraints, all linear; 452000 nonzeros
    452 equality constraints
    500 inequality constraints
1 linear objective; 219500 nonzeros.

Gurobi 8.1.0: threads=4
Gurobi 8.1.0: optimal solution; objective 97309
99480 simplex iterations
:      _varname[j]  _var[j]      :=
51      'x[1,51]'      1
501      'x[2,1]'      1
1006      'x[3,6]'      1
1601      'x[4,101]'    1
2151      'x[5,151]'    1
2552      'x[6,52]'     1
3056      'x[7,56]'     1
3502      'x[8,2]'      1
4057      'x[9,57]'     1
4606      'x[10,106]'   1
      . . .

```

Figura 3.5: Resultado layout 8 instância de 500 paletes

contrapartida, para o layout 8, são necessários, aproximadamente, 4 dias.

A figura 3.5 mostra um pequeno excerto dos resultados do layout 8. Podemos ver que obtivemos uma solução ótima, de valor 97309. Existem 226000 variáveis  $x[i,j]$ , todas binárias. Também podemos ver em qual nível deve ser colocada uma paleta. Por exemplo, a variável número 51 indica que a paleta 1 deve ser colocada no nível 51, a paleta 2 no nível 1, e assim sucessivamente.

## Capítulo 4

# Fase II - (Variante) Problema de Afetação Generalizado (Binário)

### 4.1 Descrição do problema

Na Fase II foi estudado o armazenamento das paletes, considerando que é possível colocar em cada nível (alvéolo) de cada RACK de armazenamento mais do que uma paleta, respeitando a largura disponível (L) em cada RACK. A Figura 4.1 ilustra um nível (alvéolo) de uma RACK com três paletes.

Cada largura (L) de cada RACK foi considerada fixa e igual a 3200 cm, o que permite a armazenagem de quatro europaletes (800x1200). Com esta dimensão só é possível armazenar três paletes americanas (dimensões 1000x1200).

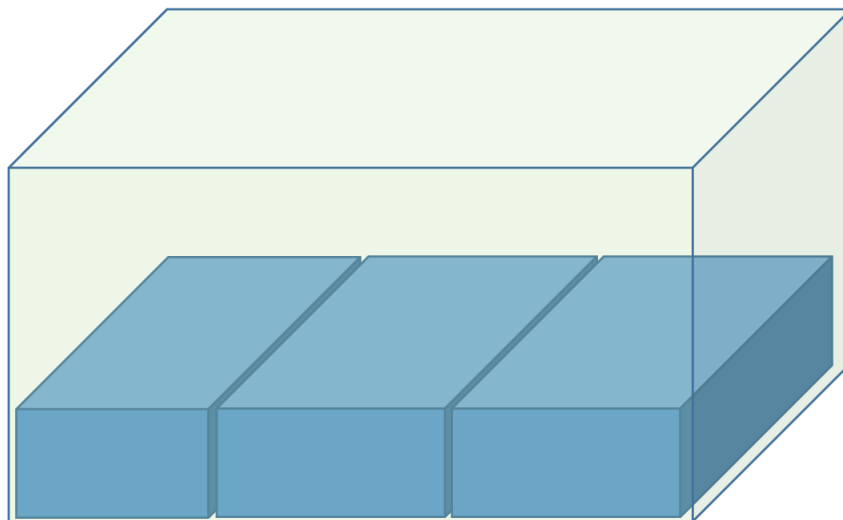


Figura 4.1: Nível de uma RACK com três paletes.

O objetivo da fase 2 é modelar e obter uma solução ótima para o tempo total de arrumação das instâncias de paletes. Ao contrario da fase 1, nesta fase em cada nível pode ser colocada mais que uma paleta.

Nesta fase, foi ignorada a altura das paletes, assumindo que a altura do nível na RACK é suficiente para armazenar qualquer paleta, isto é, cada nível tem uma altura igual ou maior que a altura da paleta mais alta. A profundidade da RACK coincide com medida Y das paletes. Os parâmetros importantes das paletes utilizados nesta fase foram a sua taxa de utilização e a sua referência.

### 4.2 ficheiro '.mod' usado na fase 2

Na figura 4.2 encontra-se o ficheiro '.mod' usado na fase 2.

A linha (1) e (3) definem os parâmetros que são lidos do ficheiro '.dat'. No ficheiro '.dat' encontra-se a instância de paletes e de níveis usados na respetiva simulação.

As linhas (5) e (6) definem que P e N correspondem, respetivamente, às paletes e níveis.

Na linha (11) definimos uma variável,  $x[i,j]$ , binária, que corresponde à colocação da paleta  $i$  no nível  $j$ .

Na linha (13) encontra-se a função objetivo, denominada 'TempoTotal'. O objetivo é minimizar o tempo total despendido na atribuição de cada paleta a cada nível, tal como na fase 1.

Nas linhas (15) e (16) correspondem às restrições que são respeitadas. Nesta fase, as restrições são ligeiramente diferentes que as colocadas na fase 1. Na linha (15), 'AfetarPaleta', indica que cada paleta tem que ser atribuída a um nível. Na linha (16), a restrição 'OcuparNivel', indica que cada nível pode conter mais que uma paleta, mas a largura máxima de cada nível é igual ou inferior ao parâmetro 'Nivel[j,2]', que no nosso projeto definimos como 3200 cm.

```
1 param N_paletes;
2
3 param N_niveis;
4
5 set P := {i in 1..N_paletes};
6 set N := {j in 1..N_niveis};
7
8 param Paleta {i in P, j in 1..4};
9 param Nivel {i in N, j in 1..2};
10
11 var x {i in P, j in N}, binary;
12
13 minimize TempoTotal : sum {i in P, j in N} Paleta[i,4] * Nivel[j,1] * x[i,j];
14
15 s.t. AfetarPaleta {i in P} : sum {j in N} x[i,j]=1;
16 s.t. OcuparNivel {j in N} : sum {i in P} Paleta[i,3] * x[i,j] <= Nivel[j,2];
```

Figura 4.2: ficheiro '.mod' usado na fase 2

### 4.3 Desenho da resolução

O objetivo da fase 2 é modelar e obter uma solução ótima que coloque todas as paletes no armazem, sendo que cada nível tem uma largura máxima de 3200 cm. Na fase 2, o procedimento foi o mesmo que o utilizado na fase 1. Simulamos três instâncias diferentes (10, 50 e 100) de paletes. Para cada instância de paletes, desenhamos três layouts diferentes de armazéns.

No anexo A, bem como em formato digital, encontra-se os nove layouts usados, dos quais os layouts 1,2 e 3 são usados para simular a instância de 10 paletes, os layouts 4,5 e 6 são usados para simular a instância de 50 paletes e os layouts 7,8 e 9 são usados para simular a instância de 500 paletes.

As instâncias de paletes para utilizadas nesta fase, foram as mesmas que utilizadas na fase anterior. Assim, podemos comparar os resultados do comportamento dos layouts para as duas fases.

### 4.4 Resultados - instância de 10 paletes

Para a instância de 10 paletes, foram concebidos três layouts diferentes de armazéns e comparamos os resultados obtidos para estes três layouts.

A instância de 10 paletes usada na simulação foi a mesma para os três layouts e pode-se encontrar em anexos.

No anexo A, nas figuras A.1, A.2 e A.3 encontra-se os layouts 1, 2 e 3, respetivamente.

Na pasta 'anexos', mais concretamente no 'path': 'anexos/resultados/fase 1/instancia 10 pale-



tes', encontra-se os resultados de cada uma das três simulações dos três layouts para a mesma instância de 10 paletes.

Para o layout 1, o valor da função objetivo foi de 388. O que podemos concluir que, para o layout 1, são precisos 388 segundos para a que a colocação de todas as paletes.

O valor obtido na fase 2 para o layout 1 é consideravelmente inferior ao valor obtido na fase 1 para o mesmo layout. Isto deve-se ao facto de que nesta fase ser possível colocar mais do que uma paleta em cada nível.

No caso do layout 2, o valor da função objetivo é de 284. Para o layout 3, o seu valor é de 388.

Os valores da função objetivo nesta fase, para os três layouts, são consideravelmente inferiores.

Na fase 1, o layout 3 tinha um valor de 731 segundos e o layout 1 um valor superior de 813. Nesta fase, os tempos do layout 1 e 3 são os mesmos, 388.

```
100 variables, all binary
20 constraints, all linear; 200 nonzeros
    10 equality constraints
    10 inequality constraints
1 linear objective; 100 nonzeros.

Gurobi 8.1.0: threads=4
Gurobi 8.1.0: optimal solution; objective 388
15 simplex iterations
: _varname[j] _var[j]      :=
1   'x[1,1]'      1
11  'x[2,1]'      1
22  'x[3,2]'      1
31  'x[4,1]'      1
41  'x[5,1]'      1
56  'x[6,6]'      1
66  'x[7,6]'      1
76  'x[8,6]'      1
86  'x[9,6]'      1
92  'x[10,2]'     1
;

:_varname[j] _var[j]      :=
;
```

Figura 4.3: resultados layout 1

```
120 variables, all binary
22 constraints, all linear; 240 nonzeros
    10 equality constraints
    12 inequality constraints
1 linear objective; 120 nonzeros.

Gurobi 8.1.0: threads=4
Gurobi 8.1.0: optimal solution; objective 388
17 simplex iterations
: _varname[j] _var[j]      :=
1   'x[1,1]'      1
13  'x[2,1]'      1
26  'x[3,2]'      1
37  'x[4,1]'      1
49  'x[5,1]'      1
64  'x[6,4]'      1
76  'x[7,4]'      1
88  'x[8,4]'      1
100 'x[9,4]'      1
110 'x[10,2]'     1
;

:_varname[j] _var[j]      :=
;|
```

Figura 4.4: resultados layout 3

Analisando os resultados dos layouts 1 e 3, representados na figura 4.3 e 4.4 respetivamente, podemos facilmente ver o porquê de a função objetivo possuir o mesmo valor. As 10 paletes são distribuídas por 3 níveis em ambos os layouts, cujo tempo de acesso a cada nível é o mesmo em cada layout.

Nos três layouts foram obtidas soluções ótimas. Admitindo que o principal critério para a escolha de um layout é o tempo que demora as paletes a serem armazenadas no armazém, concluímos que o layout 2 é o que apresenta uma melhor configuração.

## 4.5 Resultados - instância de 50 paletes

Para a instância de 50 paletes, foram concebidos três layouts diferentes de armazéns e comparados os resultados obtidos para estes três layouts.

A instância de 50 paletes usada na simulação foi a mesma para os três layouts e pode-se encontrar em anexos.

No anexo A, nas figuras A.4, A.5 e A.6 encontra-se os layouts 4, 5 e 6, respetivamente.

Do mesmo modo que para a instância de 10 paletes analisado na secção anterior, para o layout 4, o valor da função objetivo foi de 1429. O que podemos concluir que, para o layout 4, são precisos 1429 segundos para as paletes serem na totalidade armazenadas no armazém. Comparando este valor com o valor da fase 1, é consideravelmente inferior, uma vez que num mesmo nível podem ser colocadas mais que uma paleta.

No caso do layout 5, o valor da função objetivo é de 968. Para o layout 6, o seu valor é de 1026.

```
2646 variables, all binary
103 constraints, all linear; 5292 nonzeros
    49 equality constraints
    54 inequality constraints|
1 linear objective; 2646 nonzeros.

Gurobi 8.1.0: threads=4
Gurobi 8.1.0: optimal solution; objective 968
126 simplex iterations
:      _varname[j] _var[j]      :=
10      'x[1,10]'      1
64      'x[2,10]'      1
118     'x[3,10]'      1
163     'x[4,1]'       1
217     'x[5,1]'       1
271     'x[6,1]'       1
334     'x[7,10]'      1
379     'x[8,1]'       1
434     'x[9,2]'       1
490     'x[10,4]'      1
559     'x[11,19]'     1
613     'x[12,19]'     1
661     'x[13,13]'     1
713     'x[14,11]'     1
[...]
```

Figura 4.5: Resultado layout 5 instância de 50 paletes

Nesta fase, e como esperado, o valor das função objetivo para os layouts 5 e 6 é menor comparativamente com a fase 1.

O layout que possui um melhor valor na função objetivo (o que contém o menor valor), é o layout 5. Este layout é o que apresenta mais alinhamentos e um maior número de níveis em cada rack, o que leva a que o tempo de acesso a cada nível, no geral, é menor, tornando este layout o mais eficiente.

Analisando o resultado do layout 5, representado um pequeno excerto na figura 4.5, podemos ver que foi obtida uma solução ótima, de valor 968, o que indica, e admitindo que os parâmetros indicados no início são os mesmos, para colocar todas as paletes no armazém, são necessários 968 segundos. Existem 2646 variáveis  $x[i,j]$ . De um total de 54 níveis existentes no layout 5, só foram utilizados 17 níveis para a colocação das 49 paletes. Ao observar a figura 4.5 podemos ver que no nível 10, foram colocadas as paletes 1,2,3 e 7, e no nível 19, foram colocadas as paletes 11 e 12.

## 4.6 Resultados - instância de 500 paletes

Do mesmo modo que nos anteriores resultados, para a instância de 500 paletes foram concebidos três layouts diferentes de armazéns e comparados os resultados obtidos para estes três layouts.

A instância de 500 paletes usada na simulação foi a mesma para os três layouts e pode-se encontrar em anexos.

No anexo A, na figura A.7 encontra-se o layout 8. Os layouts 7 e 9, devido às suas características, só foi possível colocar em formato digital na pasta anexos.

O procedimento utilizado foi o mesmo que na fase 1. Os resultados, como esperado, foram consideravelmente diferente.

Para o layout 7, o valor da função objetivo foi de 167516. O que podemos concluir que, para o layout 7, são precisos 167516 segundos para a afetação das 500 paletes estar concluída.

No caso do layout 8, o valor da função objetivo é de 63030. Para o layout 9, o seu valor é de 94607.

```
226000 variables, all binary
952 constraints, all linear; 452000 nonzeros
      452 equality constraints
      500 inequality constraints
1 linear objective; 219500 nonzeros.

Gurobi 8.1.0: threads=4
Gurobi 8.1.0: optimal solution; objective 63030
157148 simplex iterations
1 branch-and-cut nodes
:      _varname[j]  _var[j]      :=
1      'x[1,1]'      1
501    'x[2,1]'      1
1001   'x[3,1]'      1
1551   'x[4,51]'     1
2051   'x[5,51]'     1
2551   'x[6,51]'     1
3101   'x[7,101]'    1
3651   'x[8,151]'    1
4002   'x[9,2]'      1
4506   'x[10,6]'     1
5151   'x[11,151]'   1
5502   'x[12,2]'     1
6151   'x[13,151]'   1
6552   'x[14,52]'    1
7056   'x[15,56]'    1
7601   'x[16,101]'   1
8052   'x[17,52]'    1
8601   'x[18,101]'   1
9006   'x[19,6]'     1
9556   'x[20,56]'    1
10002  'x[21,2]'     1
```

Figura 4.6: Resultado layout 8 instância de 500 paletes

Por comparação com a fase 1, os valores da função objetivos para os layouts 7, 8 e 9 na fase 2 são consideravelmente melhores, o que era de esperar. Devido ao layout 7 possuir apenas um único alinhamento com 100 racks, foi o layout que apresentou piores resultados. O layout 9, por possuir mais um alinhamento que o layout 7, diminuindo assim o tempo de acesso, no geral, aos níveis de cada rack, apresentou melhores resultados que o layout 7.

Seguindo o mesmo ideal, o layout 8, que possui mais alinhamentos que os layouts 7 e 9, é o que apresenta melhores resultados. Na figura 4.6 podemos ver a vista superior do layout 8. Possui 10 alinhamentos, 5 corredores e em cada alinhamento possui 10 racks.

	alinhamento 1	Rack 1	Rack 2	Rack 3	Rack 4	Rack 5	Rack 6	Rack 7	Rack 8	Rack 9	Rack 10
porta 1		0	1	2	3	4	5	6	7	8	9
		0	1	2	3	4	5	6	7	8	9
	alinhamento 2	Rack 11	Rack 12	Rack 13	Rack 14	Rack 15	Rack 16	Rack 17	Rack 18	Rack 19	Rack 20
	alinhamento 3	Rack 21	Rack 22	Rack 23	Rack 24	Rack 25	Rack 26	Rack 27	Rack 28	Rack 29	Rack 30
		1	2	3	4	5	6	7	8	9	10
		1	2	3	4	5	6	7	8	9	10
	alinhamento 4	Rack 31	Rack 32	Rack 33	Rack 34	Rack 35	Rack 36	Rack 37	Rack 38	Rack 39	Rack 40
	alinhamento 5	Rack 41	Rack 42	Rack 43	Rack 44	Rack 45	Rack 46	Rack 47	Rack 48	Rack 49	Rack 50
		2	3	4	5	6	7	8	9	10	11
		2	3	4	5	6	7	8	9	10	11
	alinhamento 6	Rack 51	Rack 52	Rack 53	Rack 54	Rack 55	Rack 56	Rack 57	Rack 58	Rack 59	Rack 60
	alinhamento 7	Rack 61	Rack 62	Rack 63	Rack 64	Rack 65	Rack 66	Rack 67	Rack 68	Rack 69	Rack 70
		3	4	5	6	7	8	9	10	11	12
		3	4	5	6	7	8	9	10	11	12
	alinhamento 8	Rack 71	Rack 72	Rack 73	Rack 74	Rack 75	Rack 76	Rack 77	Rack 78	Rack 79	Rack 80
	alinhamento 9	Rack 81	Rack 82	Rack 83	Rack 84	Rack 85	Rack 86	Rack 87	Rack 88	Rack 89	Rack 90
		4	5	6	7	8	9	10	11	12	13
		4	5	6	7	8	9	10	11	12	13
	alinhamento 10	Rack 91	Rack 92	Rack 93	Rack 94	Rack 95	Rack 96	Rack 97	Rack 98	Rack 99	Rack 100

Figura 4.7: vista superior do layout 8

Na figura 4.5 podemos ver que foi encontrada uma solução ótima para a afetação generalizada das paletes.

Do valor da solução ótima, podemos concluir que, e assumindo que cada dia útil de trabalho é composto por oito horas, serão precisos 2 dias e 3 horas, aproximadamente, para as 500 paletes estarem armazenadas no armazém. Na fase um, se cada nível apenas fosse permitido colocar uma paleta, seriam precisos 3 dias e 4 horas. Se considerássemos o layout 7 na fase 2 seriam necessários aproximadamente 6 dias e na fase 1 aproximadamente 15 dias. Podemos concluir que o layout 7 é desaconselhado em detrimento do layout 8.

Da figura 4.6 podemos ver que para um mesmo nível foram atribuídas mais que uma paleta. Os níveis de racks que se encontram mais para o fim de cada alinhamento encontram-se vazios, o que é intuitivo, uma vez que possuem um tempo de acesso maior que os níveis de racks de alinhamentos mais próximos da porta.

## Capítulo 5

# Fase III - Problema de Afetação e Layout de Armazém

### 5.1 Descrição do problema

Na Fase III pretendeu-se estudar o armazenamento das paletes, considerando que é possível colocar em cada nível (alvéolo) de cada RACK de armazenamento mais do que uma paleta, respeitando a largura disponível ( $L$ ) em cada RACK. A Figura 9 ilustra um nível (alvéolo) de uma RACK com três paletes.

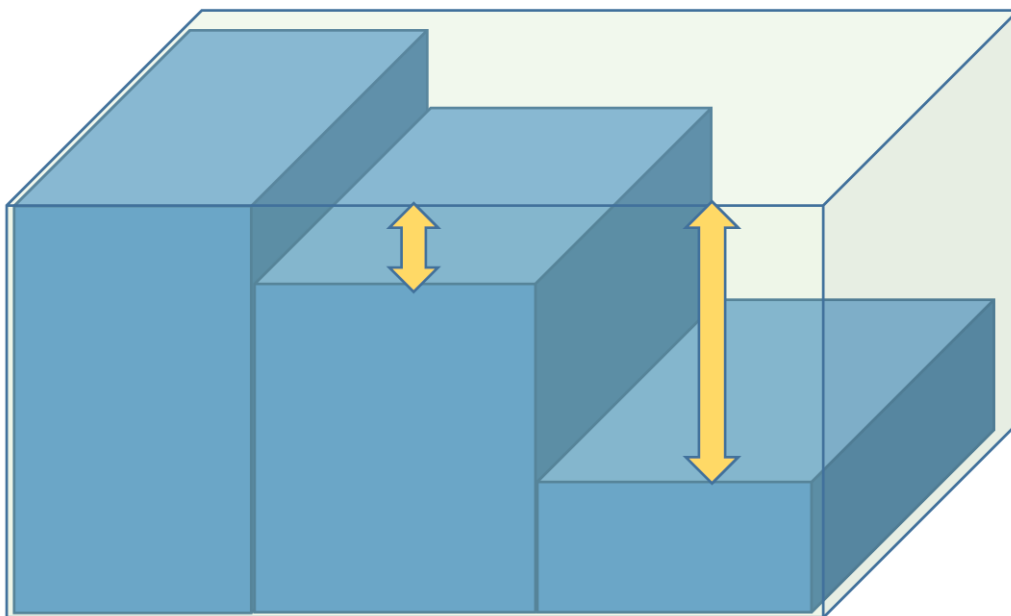


Figura 5.1: Nível de uma RACK com três paletes.

O objetivo é minimizar o espaço vazio por cima de cada paleta. O nivelamento da altura das paletes dispostas no mesmo nível (alvéolo) de cada RACK permitirá ocupar menor volumetria global do armazenamento.

Nesta fase, pretendia-se reajustar a altura de cada nível.

Esta fase foi a fase mais desafiante. O ficheiro '.mod' foi mais complexo e os dados mais aproximados com a realidade.

### 5.2 ficheiro '.mod' usado na fase 3

Nesta fase, o ficheiro '.mod' foi sofreu modificações significativas. O ficheiro '.mod' usado nesta fase encontra-se no anexo e na figura 5.2.

```

1 param N_paletes;
2 param N_niveis;
3 param N_rack; # numero de racks usadas
4 param HMaxRack; # altura maxima de uma rack
5
6 set P := {i in 1..N_paletes}; # P representa Paletes
7 set N := {j in 1..N_niveis}; # N representa niveis
8 set R := {k in 1..N_rack};
9
10 param Palette {i in P, j in 1..4}; # Referência → Altura → Largura → Taxa de Utilização
11 param Nivel {i in N, j in 1..3}; # Acessibilidade → Largura → n° rack
12
13 var x {i in P, j in N}, binary;
14 var Hr {k in R} >= 0;
15
16 # alt corresponde a altura de cada nível
17 var alt {j in N}, >= 0;
18 # des corresponde ao desperdício de cada nível
19 var des {i in P}, >= 0;
20
21 var auxFO1, >= 0;
22 var auxFO2, >= 0;
23
24 #FO tempo total
25 minimize TempoTotal : sum {i in P, j in N} Palette[i,4] * Nivel[j,1] * x[i,j];
26 #FO desperdicio
27 #minimize ValorDesperdi : sum {i in P} des[i];
28
29 s.t. AfetarPaleta {i in P} : sum {j in N} x[i,j] = 1;
30 s.t. OcuparNivel {j in N} : sum {i in P} Palette[i,3] * x[i,j] <= Nivel[j,2];
31 s.t. RestricaoAlt {i in P, j in N} : alt[j] >= Palette[i,2] * x[i,j];
32 s.t. Desperdi {i in P, j in N} : x[i,j] = 1 ==> des[i] = alt[j] - Palette[i,2];
33
34 s.t. calcFO1 : auxFO1 = sum {i in P, j in N} Palette[i,4] * Nivel[j,1] * x[i,j];
35 s.t. calcFO2 : auxFO2 = sum {i in P} des[i];
36
37 s.t. AlturaRack {k in R} : Hr[k] = sum {j in N} Nivel[j,3] = k * alt[j];
38 s.t. AlturaRack1 {k in R} : Hr[k] <= HMaxRack;
39
40 #s.t. calcMO : auxFO1 <= 1000;

```

Figura 5.2: ficheiro .mod usado na fase 3

As linhas (1),(2),(3) e (4) são quatro parâmetros que recebem do ficheiro de dados o número de paletes usadas, o número de níveis, o número de racks existentes no armazém e a altura máxima de cada rack suportada pelo armazém, respetivamente.

As linhas (6) a (8) definimos que P representa as paletes, N representa os níveis e R as racks.

Na linha (10) definimos que o parâmetro Palette que possui quatro 'campos': ( Referência, altura, largura e taxa de utilização). Na linha (11) definimos que o parâmetro Nivel tem três 'campos': (Acessibilidade, largura e n° da rack).

Na linha (13) e (14) definimos duas variáveis. A variável  $x[i,j]$  binária, que representa a existência ou não de uma paleta (i), num determinado nível (j). A variável  $Hr(k)$  representa a altura de cada rack. Na linha (17) definimos uma variável que representa a altura de cada nível, e na linha (19) o desperdício associado a cada paleta.

A linha (21) está definida uma variável onde será armazenada o valor da função objetivo 1, a responsável por minimizar o tempo total na arrumação das paletes. Na linha (22) a variável definida tem

como objetivo armazenar o valor da função objetivo 2, responsável pela minimização do desperdício associado às paletes.

As linhas (25) e (27) são responsáveis, respetivamente, pelas funções objetivos, cuja finalidades são minimizar o tempo total e o desperdício total. Só uma pode estar ativa em cada compilação.

As linhas 29 a 40 é onde estão declaradas as restrições associadas às funções objetivos. A linha (29) é responsável pela afetação das paletes. A linha (30) indica que a largura máxima das paletes colocadas em cada nível não deve exceder a largura do respetivo nível. A linha (31) indica que a altura de cada nível tem que ser maior ou igual à altura da paleta mais alta colocada nesse nível. A linha (32) é responsável pelo calculo do desperdício total. O desperdício em cada nível é igual à soma da diferença entre a altura de cada nível e a altura de cada paleta inserida no respetivo nível.

As linhas (34) e (35) são responsáveis pelo calculo dos valores das variáveis 'calcFO1' e 'calcFO2' definidas anteriormente.

A linha (37) é responsável pela altura de cada rack. E, por fim, a linha (38) é a restrição responsável por cada rack não exceder a altura máxima definida para o nosso armazém.

A linha (40) tem como finalidade restringir o valor da função objetivo 1, para encontrar-mos pontos intermédios entre o tempo total da arrumação das paletes, e o desperdício total.

### 5.3 Desenho da resolução

A fase 3 foi a fase mais complexa, a que mais trabalho foi dedicado.

Tal como nas anteriores fases, simulamos a fase 3 para cada instância de paletes (50, 100 e 500). Para cada instância, simulamos três layouts de armazéns diferentes.

Nesta fase existem dois objetivos: minimizar o tempo total na distribuição das paletes pelo armazém e, diminuir o desperdício entre cada nível e as paletes. Para tal, foram simuladas duas funções objetivo: uma responsável pela diminuição dos tempos na afetação das paletes (FO1) e a segunda responsável por minimizar o desperdício das paletes (FO2).

Em cada simulação só uma função objetivo pode estar ativa. Ao simularmos apenas com uma função objetivo ativa, obtivemos as soluções ótimas em cada caso. Como a solução ótima da função objetivo 1 corresponderia a uma solução com um valor elevado no desperdício das paletes, e vice-versa, para obter soluções intermédias nas duas funções objetivos, fomos restringindo o valor da função objetivo 1.

Para cada instância de paletes, escolhemos um layout de um armazém para colocar os dados das duas funções objetivos num gráfico, no eixo das abcissas colocamos a FO1 e no eixo das ordenadas a FO2, e comparamos os valores obtidos.

Nesta fase, o número de níveis em cada rack era dinâmico e calculado pelo programa. A altura máxima definida para as racks é de 6 metros (600 cm).

No anexo A, bem como em formato digital, encontra-se os nove layouts usados, dos quais os layouts 1,2 e 3 são usados para simular a instância de 10 paletes, os layouts 4,5 e 6 são usados para simular a instância de 50 paletes e os layouts 7,8 e 9 são usados para simular a instância de 500 paletes.

As instâncias de paletes utilizadas nesta fase, foram as mesmas que utilizadas na fase anterior. Assim, podemos comparar os resultados do comportamento dos layouts para as três fases.

### 5.4 Resultados - instância de 10 paletes

Para a instância de 10 paletes, foram concebidos três layouts diferentes de armazéns e comparamos os resultados obtidos para estes três layouts.

A instância de 10 paletes usada na simulação foi a mesma para os três layouts e pode-se encontrar em anexos.

No anexo A, nas figuras A.1, A.2 e A.3 encontra-se os layouts 1, 2 e 3, respetivamente.

Para o layout 1, e só com a função objetivo responsável por minimizar o tempo na distribuição das paletes pelo armazém (FO1), o seu valor foi de 388. Valor igual ao obtido na fase 2. O desperdício

das paletes associado foi de 316. O que podemos concluir que para a colocação das 10 paletes no armazém, são necessários 388 segundos, havendo um desperdício de 316 cm entre os níveis e as paletes.

Se ativarmos apenas a função objetivo FO2, responsável por minimizar o desperdício das paletes, o valor da FO1 foi de 707 e o valor de FO2, naturalmente, foi zero. O que indica que uma solução ótima para o desperdício das paletes corresponde a um tempo de afetação das paletes de 707 segundos, um valor bastante superior obtido ao ativarmos apenas a FO1.

Se restringirmos o valor de FO1 para menor ou igual a 400 e ativarmos apenas FO2, em FO1 obtivemos 392 e em FO2 52, o que não sendo uma solução ótima em cada caso, poderá ser uma solução que pode satisfazer mais clientes.

Com o mesmo procedimento, obtivemos os seguintes resultados para o layout 3:

- apenas FO1 ativa:
  - $FO1 = 388$
  - $FO2 = 82$
- apenas FO2 ativa:
  - $FO1 = 790$
  - $FO2 = 0$
- apenas FO2 ativa mas FO1 menor ou igual a 400:
  - $FO1 = 392$
  - $FO2 = 52$

Uma das conclusões a tirar entre o layout 1 e 3, e se o nosso único objetivo for minimizar o tempo total da afetação das paletes, o layout 3 possui um desperdício menor que o layout 1.

Para o layout 2, e como nas outras fases foi o que possui melhores resultados, teve uma análise mais detalhada, sendo estes alguns resultados obtidos:

- apenas FO1 ativa:
  - $FO1 = 284$
  - $FO2 = 136$
- apenas FO2 ativa:
  - $FO1 = 615$
  - $FO2 = 0$
- apenas FO2 ativa mas FO1 menor ou igual a 400:
  - $FO1 = 375$
  - $FO2 = 0$
- apenas FO2 ativa mas FO1 menor ou igual a 300:
  - $FO1 = 292$
  - $FO2 = 78$
- apenas FO2 ativa mas FO1 menor ou igual a 284:
  - $FO1 = 284$
  - $FO2 = 96$
- apenas FO2 ativa mas FO1 menor ou igual a 345:



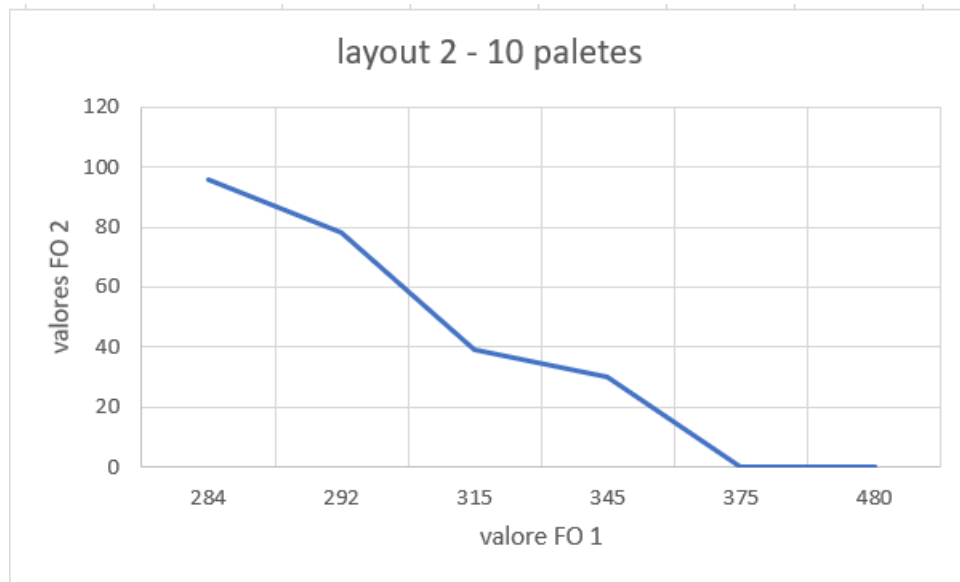


Figura 5.3: dados layout 2

- FO1 = 345
- FO2 = 39

Através dos dados recolhidos, foi possível construir o gráfico da figura 5.3.

O valor da FO1, estando apenas esta ativa é de 284, correspondendo a um desperdício de 284.

Se apenas o desperdício for minimizado, o valor de FO2 é zero e o de FO1, inicialmente foi 615. Mas se limitar-mos o valor de FO1 a menor do que 400, o desperdício continuo a ser zero. O valor mínimo que conseguimos, para um valor de desperdício não nulo, foi de 375 para a FO1.

Neste caso, a solução ótima obtida em FO1 e a solução ótima obtida em FO2, podem não satisfazer as necessidades de quem procura um armazém eficaz. Uma boa solução pode ser um valor mediano entre as duas funções objetivo, visto que, ao descermos no tempo total da afetação das paletes, consequentemente estamos a aumentar no despercebo do armazém.

Os dados da figura 5.4 foram obtidos quando ativamos apenas a FO2. Basicamente, o que se sucedeu, é que as paletes com alturas iguais ficaram todas no mesmo nível, as que possuem alturas diferentes foram colocadas noutros níveis, havendo um desperdício nulo.

Nesta fase também podemos obter valores tais como: qual deve ser a altura de cada rack (neste exemplo, a rack 1 deve ter atura 2,67m e a rack 2, 270m),e qual deve ser a altura de cada nível. Estes parâmetros ajudam a construir um armazém otimizado.

```

Gurobi 8.1.0: threads=4
Gurobi 8.1.0: optimal solution; objective 0
191 simplex iterations
47 branch-and-cut nodes
plus 1 simplex iteration for intbasis
: _varname[j] _var[j]      :=
1   'x[1,1]'      1
20  'x[2,10]'     1
22  'x[3,2]'      1
32  'x[4,2]'      1
42  'x[5,2]'      1
52  'x[6,2]'      1
68  'x[7,8]'      1
76  'x[8,6]'      1
83  'x[9,3]'      1
99  'x[10,9]'     1
;

: _varname[j] _var[j]      :=
101 'Hr[1]'       267
102 'Hr[2]'       270
103 'alt[1]'      109
104 'alt[2]'      79
105 'alt[3]'      79
108 'alt[6]'      61
110 'alt[8]'      61
111 'alt[9]'      39
112 'alt[10]'     109
123 auxFO1       615
;

```

Figura 5.4: dados layout 2

## 5.5 Resultados - instância de 50 paletes

Para a instância de 50 paletes, foram concebidos três layouts diferentes de armazéns e comparamos os resultados obtidos para estes três layouts, tal como nas fases anteriores.

A instância de 50 paletes usada na simulação foi a mesma para os três layouts e pode-se encontrar em anexos.

No anexo A, nas figuras A.4, A.5 e A.6 encontra-se os layouts 4, 5 e 6, respetivamente.

Para esta instância foram analisados os dados das funções objetivos FO1 e FO2. (FO1 - minimizar o tempo total da colocação das paletes no armazém e FO2 minimizar o desperdício entre as paletes e os níveis)

Para o layout 4, estes foram os principais resultados obtidos:

- apenas FO1 ativa:
  - FO1 = 1429
  - FO2 = 606
- apenas FO2 ativa:
  - FO1 = 3753
  - FO2 = 0
- apenas FO2 ativa mas FO1 menor ou igual a 1429:

- FO1 = 1429
- FO2 = 414
- apenas FO2 ativa mas FO1 menor ou igual a 2000:
  - FO1 = 1992
  - FO2 = 59
- apenas FO2 ativa mas FO1 menor ou igual a 2471:
  - FO1 = 2471
  - FO2 = 0

Para o layout 6, estes foram os principais resultados obtidos:

- apenas FO1 ativa:
  - FO1 = 1026
  - FO2 = 286
- apenas FO2 ativa:
  - FO1 = 3753
  - FO2 = 0
- apenas FO2 ativa mas FO1 menor ou igual a 1429:
  - FO1 = 1026
  - FO2 = 246
- apenas FO2 ativa mas FO1 menor ou igual a 2000:
  - FO1 = 1992
  - FO2 = 59
- apenas FO2 ativa mas FO1 menor ou igual a 2471:
  - FO1 = 2296
  - FO2 = 0

Através dos dados, construímos o gráfico da figura 5.5, no qual concluímos que o layout 6 é descartado pelo layout 4, uma vez que não é possível encontrar uma solução com melhores resultados no layout 4 em comparação com o layout 6.

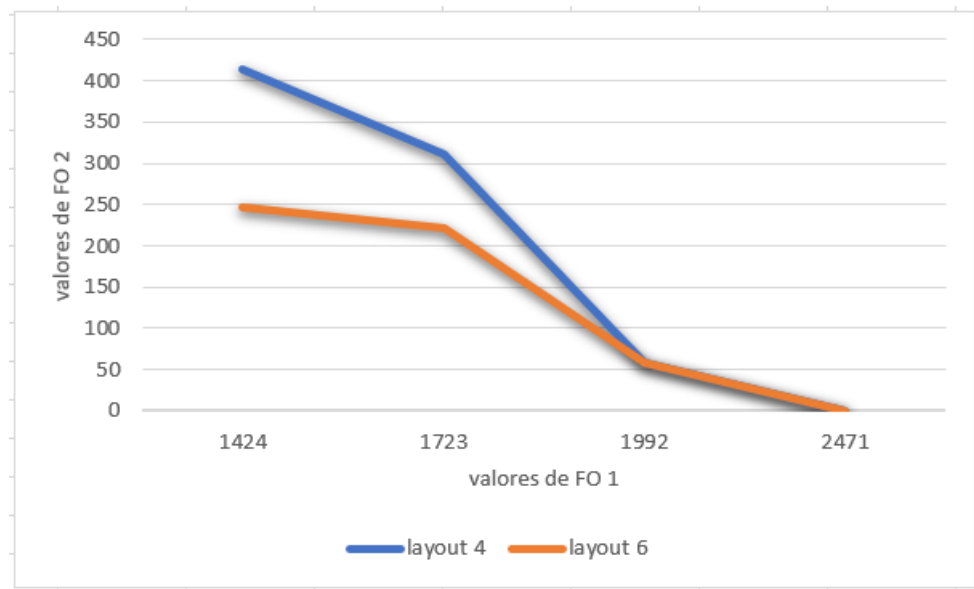


Figura 5.5: layout 4 e 6

Para o layout 5, e como este layout nas fases anteriores apresentou melhores resultados, decidimos explorar os resultados um pouco mais.

Quando a FO1 é apenas ativa, o seu valor foi de 968, com um desperdício de 656. Mas, se forçarmos a que a FO1 seja menor ou igual a 968 e ativarmos apenas FO2, o desperdício desce para 136. Isto deve-se a que, no primeiro caso, o neoServer deu uma solução ótima, mas não se preocupou a minimizar o desperdício.

Se ativarmos apenas FO2, seu valor é nulo e o valor de FO1 é de 2471. O menor valor de desperdício que conseguimos obter foi quatro, para um valor de FO1 de 998.

Se forçarmos a FO1 ser menor do 980, obtivemos para FO2 96. Através deste método, obtivemos o gráfico da figura 5.6 para a relação entre FO1 e FO2 para o layout 5

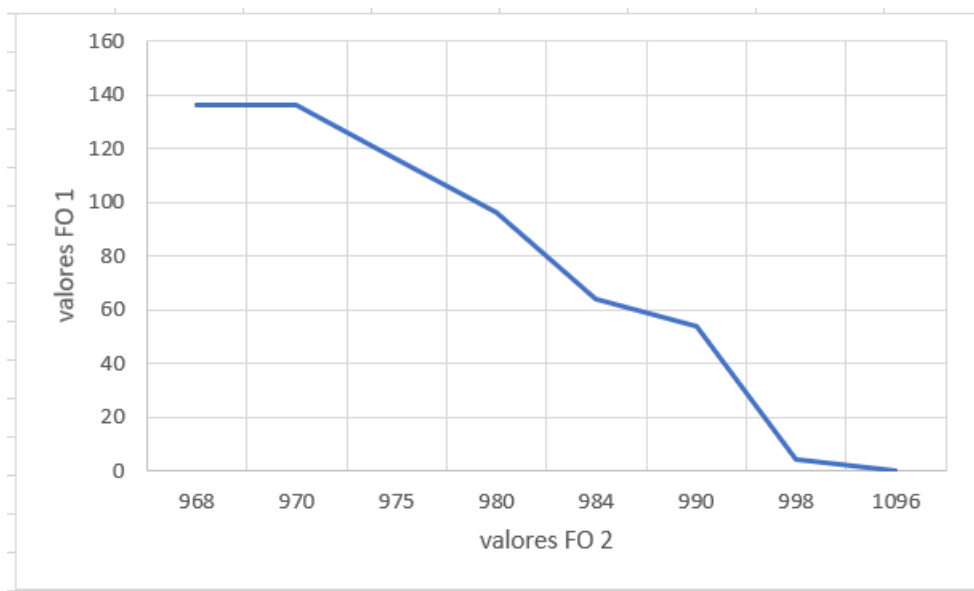


Figura 5.6: relação entre FO1 e FO2 layout 5

Através da análise do gráfico, são várias as soluções que podemos obter para as diversas necessidade. Se pretendermos um desperdício mínimo, este irá ser sempre acompanhado por um valor elevado de tempo na afetação das paletes. Se o tempo na afetação das paletes for o mais importante, a consequência a um menor tempo da afetação das paletes é um desperdício elevado de espaço entre as paletes e o nível superior.

Uma boa solução que minimiza as consequências das duas funções objetivos, ó o ponto do gráfico que está a uma menor distancia da origem (zero). Por exemplo, para o ponto (984,64), que no contexto do problema, para uma afetação das paletes no armazém em 964 segundos, corresponde um desperdício de 64 cm.

Na figura 6.6 está representado um pequeno excerto dos resultados obtidos numa simulação. Através dos resultados são vários os dados, interessantes, que podemos extrair. Por exemplo, a paleta 36 deve ir para o nível 14. No nível 24 existe um desperdício de, aproximadamente 2 cm, e a altura da rack 1 é de 267 cm.

```

1904 'x[36,14]' 1
1955 'x[37,11]' 1
2001 'x[38,3]' 1
2056 'x[39,4]' 1
2126 'x[40,20]' 1
2163 'x[41,3]' 1
2219 'x[42,5]' 1
2284 'x[43,16]' 1
2334 'x[44,12]' 1
2413 'x[45,37]' 1
2435 'x[46,5]' 1
2504 'x[47,20]' 1
2545 'x[48,7]' 1
2638 'x[49,46]' 1
2742 'des[24]' 1.99967
;

: _varname[j] _var[j] :=
2647 'Hr[1]' 267
2648 'Hr[2]' 277
2649 'Hr[3]' 197
2650 'Hr[4]' 197
2651 'Hr[5]' 257
2652 'Hr[6]' 257
2653 'Hr[7]' 219
2654 'Hr[8]' 257
2655 'Hr[9]' 187
2656 'Hr[10]' 187

```

Figura 5.7: relação entre FO1 e FO2 layout 5

## 5.6 Resultados - instância de 500 paletes

Para a instância de 500 paletes, foram concebidos três layouts diferentes de armazéns e comparamos os resultados obtidos para estes três layouts.

A instância de 500 paletes usada na simulação foi a mesma para os três layouts e pode-se encontrar em anexos.

No anexo A, na figura A.7 encontra-se o layout 8. Os layouts 7 e 9, devido às suas configurações, apenas estão representados em suporte digital, na pasta 'anexos'.

Devido à enorme quantidade de combinações de paletes e níveis, não foi possível obter todos os dados como nas instâncias anteriores.

Em relação ao layout 8, o layout que nas fases e instâncias anteriores foi o que obteve melhores resultados, os resultados que conseguimos extrair foram os seguintes:

- apenas FO1 ativa:

- FO1 = 63030
- FO2 = 27513

Se o objetivo for apenas otimizar o tempo na colocação das paletes em cada nível, serão precisos 63 030 segundos, o mesmo resultado obtido na fase 2 para o layout 8, mas, o desperdício associado será de 27513 cm, ou seja, e como cada rack tem 600 cm de altura, irá haver um desperdício de, aproximadamente, 46 racks. O que é algo que num armazém real não pode acontecer. Logo, podemos concluir a solução ótima na afetação das paletes, terá como consequência um desperdício de espaços bastante grande.

- apenas FO2 ativa mas FO1 limitada a 75000:

- FO1 = 74991
- FO2 = 0

- apenas FO2 ativa:

- FO1 = 145258
- FO2 = 0

Se apenas a função objetivo que minimiza os desperdícios estar ativa, o desperdício será otimizado ao máximo, é encontrada uma solução ótima, mas a consequência é que a afetação das paletes irá ser de 145258 segundos, aproximadamente 2,5 vezes maior que o anterior. Ao analisar os dados, vimos que na prática cada paleta irá ser colocada em cada nível, aproximadamente, resultando num desperdício em cada nível nulo como os dados da figura 5.8 mostram.

4181	'x[9,181]'	1
4612	'x[10,112]'	1
5225	'x[11,225]'	1
5663	'x[12,163]'	1
6037	'x[13,37]'	1
6765	'x[14,265]'	1
7361	'x[15,361]'	1
7676	'x[16,176]'	1
8410	'x[17,410]'	1
8991	'x[18,491]'	1
9471	'x[19,471]'	1
9868	'x[20,368]'	1
10492	'x[21,492]'	1
10558	'x[22,58]'	1
11185	'x[23,185]'	1
11901	'x[24,401]'	1
12096	'x[25,96]'	1
12222	'x[26,1222]'	1

Figura 5.8: layout 8 - 500 paletes

Devido à quantidade excessiva de combinações que podem existir entre paletes e níveis, ao tentar encontrar soluções intermédias para as duas funções objetivos, deparamos com o problema da imagem 5.9.

Presolve eliminates 101 constraints.

Adjusted problem:

227054 variables:

226952 nonlinear variables

102 linear variables

227054 algebraic constraints, all linear; 1124554 nonzeros

554 equality constraints

226500 inequality constraints

226000 logical constraints

1 linear objective; 3 nonzeros.

Gurobi 8.1.0: threads=4

ERROR: Your job was terminated because it exceeded the maximum allotted time for a job.

Please refer to the NEOS Server FAQ (<https://neos-guide.org/content/FAQ>) for more informat

Figura 5.9: limite de 3GB atingidos

Devido às enormes combinações, foi atingido o limite máximo de 3 GB de RAM no neoServer. Para tentar solucionar este problema, no ficheiro 'run' colocamos as instruções da figura 5.10 e corremos os nossos ficheiros no CPLEX em vez do Gurobi. Com um limite máximo de 15GB de RAM, também não conseguimos obter resultados.

```
#set workmem 128
#set mip strategy file 2
#set mip limits treememory 15000
```

Figura 5.10: ficheiro run - CPLEX

Como só conseguimos valores otimizados para as duas funções objetivos, não foi possível fazer os gráficos que correlacionam estas duas. A conclusão que podemos tirar, é que, para a solução ótima na afetação das paletes tem como consequência um elevado desperdício, e, para a solução ótima na otimização do desperdício, tem como consequência um elevado tempo na afetação das paletes. Deste modo, uma solução intermédia para as duas funções objetivos pode corresponder às necessidades da maioria das pessoas.



## Capítulo 6

# Conclusão

Através da realização deste projeto proposto no âmbito da unidade curricular de investigação operacional do curso de MIETI, conseguimos compreender e aplicar os conhecimentos adquiridos nas aulas e adquirir ferramentas alusivas à área da investigação operacional. O objetivos propostos na introdução foram, na generalidade, atingidos. Adquirimos competências em equipa, e aplicamos os conceitos adquiridos nas salas de aula e com trabalho autónomo.

A elaboração do presente projeto foi enriquecedor. Ficamos a conhecer alguns dos principais conceitos que regem a investigação operacional, e a importância desta na resolução de problemas.

A terceira fase do projeto, foi a fase que mais desafios envolveu. Foi nesta fase que trabalhos com a otimização dos tempos da afetação das paletes, na otimização dos desperdícios das paletes, na altura de uma rack, na altura de cada nível e no desperdício associado a cada nível, em qual nível deve ser colocada uma paleta.

As aulas práticas e teóricas tiveram um papel importante, onde aproveitamos para tirar dúvidas e debater alguns pontos importantes no desenrolar do projeto, em grupo.

Este projeto também nos possibilitou o conhecimento de várias ferramentas, nomeadamente, a linguagem AMPL, o gusek, neoServer, vários pacotes de otimização, como o 'Gurobi' e o 'CPLEX'. É de realçar o papel que a universidade do Minho tem na área da otimização numérica, juntamente com algumas universidade norte americanas em disponibilizar 'solvers', nomeadamente o 'neoServer'.

Para a elaboração deste relatório, utilizamos o software  $\text{\LaTeX}$ . Tal como dito na introdução, concluímos que este software é uma excelente forma de produzir trabalhos científicos, relatórios e textos académicos com excelente qualidade.

## Capítulo 7

# Bibliografia

1. (June 2016) Pallet racking use and maintenance guide
2. Slides colocados na BlackBoard Learn da unidade curricular
3. <https://neos-server.org/neos/index.html>
4. <http://gusek.sourceforge.net/gusek.html>
5. <https://ampl.com/products/solvers/solvers-we-sell/cplex/>
6. <https://www.latex-project.org/>

## Apêndice A

# Layouts dos armazéns

Lista-se a seguir os layouts utilizados neste projeto.

A.1 Instância de 10 paletes

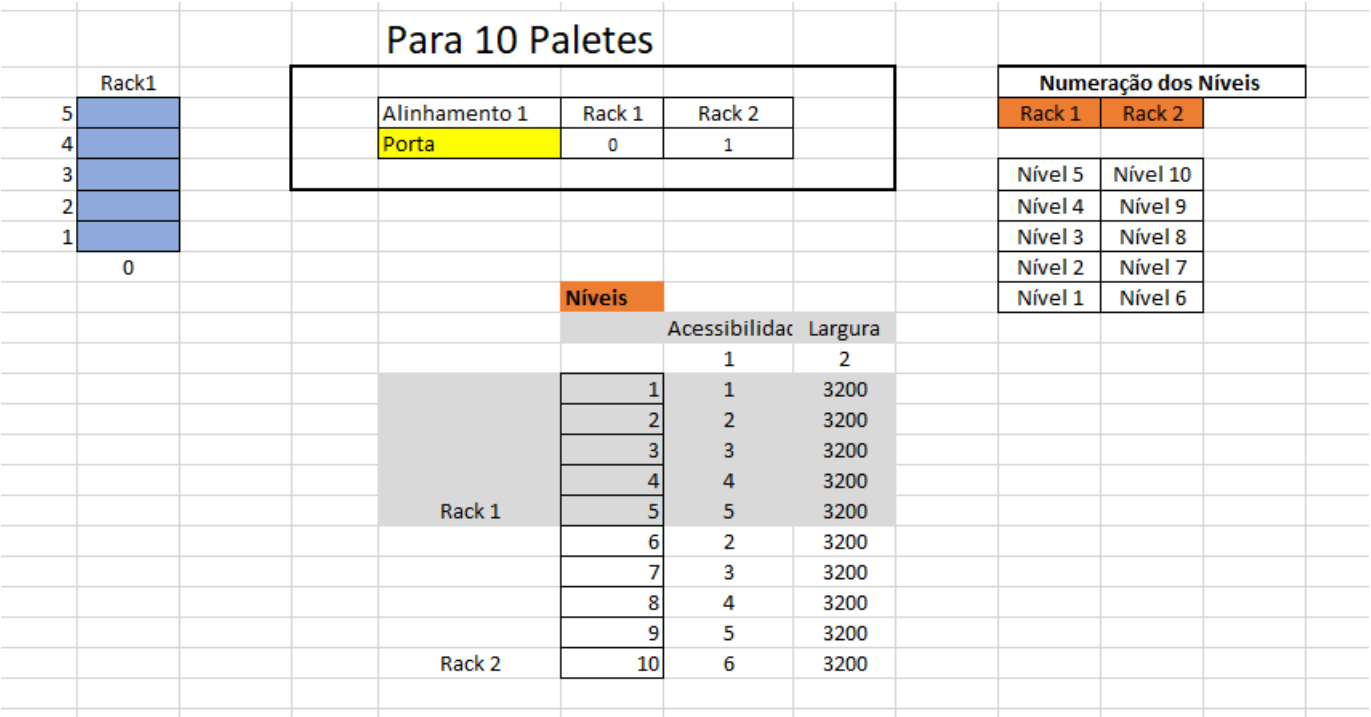


Figura A.1: layout 1 - instância de 10 paletes

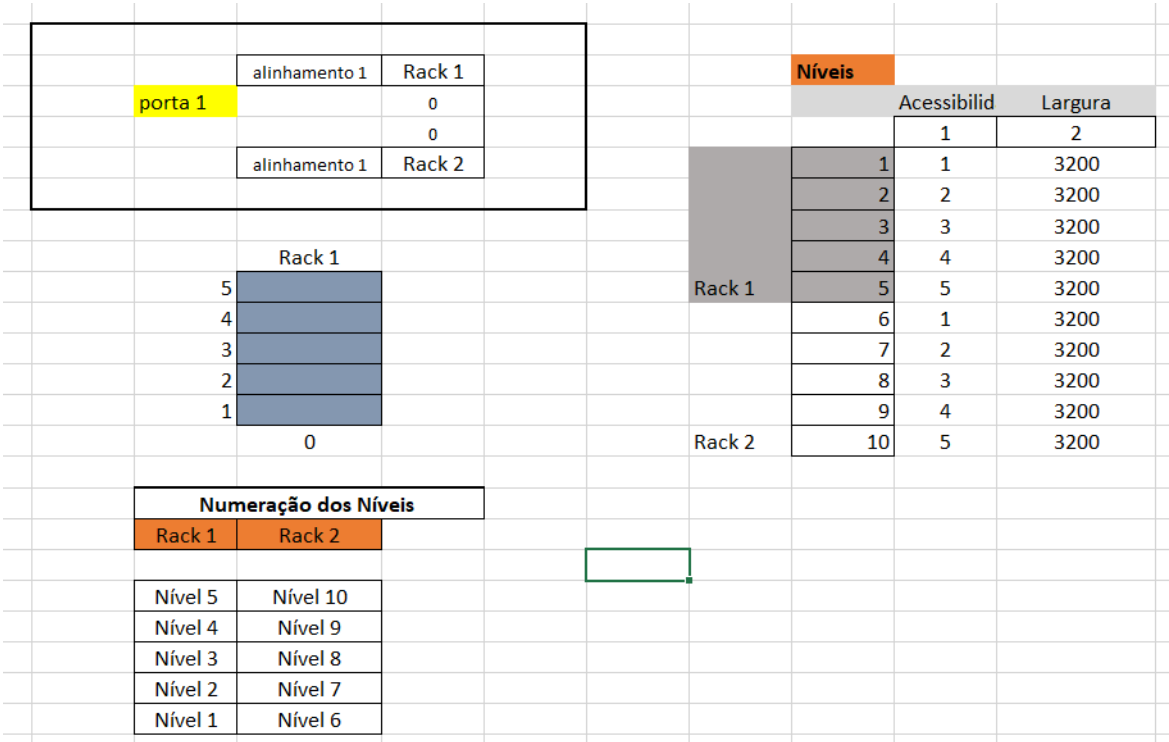


Figura A.2: layout 2 - instância de 10 paletes

	rack 1	rack 2	rack 3	rack 4		Níveis		
Porta 1	0	1	2	3			Acessibilid	Largura
							1	2
					Rack 1	1	1	3200
						2	2	3200
						3	3	3200
rack 1						4	2	3200
3						5	3	3200
2					Rack 2	6	4	3200
1					Rack 3	7	3	3200
						8	4	3200
0						9	5	3200
						10	4	3200
						11	5	3200
					Rack 4	12	6	3200

Figura A.3: layout 3 - instância de 10 paletes

## A.2 Instância de 50 paletes

Alinhamen	Rack 1	Rack 2	Rack 3	Rack 4	Rack 5	Rack 6	Rack 7	Rack 8	Rack 9	Rack 10
Porta	0	1	2	3	4	5	6	7	8	9

Rack1

5	
4	
3	
2	
1	
0	

Figura A.4: layout 4 - instância de 10 paletes

	alinhamento 1	Rack 1	Rack 2	Rack 3
porta 1		0	1	2
		0	1	2
	alinhamento 2	Rack 4	Rack 5	Rack 6
	alinhamento 3	Rack 7	Rack 8	Rack 9
		1	2	3
		1	2	3
	alinhamento 4	Rack 10	Rack 11	Rack 12
	alinhamento 5	Rack 13	Rack 14	Rack 15
		2	3	4
		2	3	4
	alinhamento 6	Rack 16	Rack 17	Rack 18
		Rack 15		
	3	Nível 3		
	2	Nível 2		
	1	Nível 1		
		5		

Figura A.5: layout 5 - instância de 10 paletes

	rack 1	rack 2	rack 3	rack 4	rack 5
	0	1	2	3	4
Porta 1					
	rack 6	rack 7	rack 8	rack 9	rack 10
	0	1	2	3	4

	rack 1				
5					
4					
3					
2					
1					
	0				

Figura A.6: layout 6 - instância de 10 paletes

### A.3 Instância de 500 paletes

Devido à enorme quantidade de racks em cada alinhamento, foi impossível colocar aqui as imagens dos layouts dos armazéns 7 e 9. Os correspondentes layouts estão na pasta 'anexos' que acompanhou este projeto no formato digital.

	alinhamento 1	Rack 1	Rack 2	Rack 3	Rack 4	Rack 5	Rack 6	Rack 7	Rack 8	Rack 9	Rack 10
porta 1		0	1	2	3	4	5	6	7	8	9
		0	1	2	3	4	5	6	7	8	9
	alinhamento 2	Rack 11	Rack 12	Rack 13	Rack 14	Rack 15	Rack 16	Rack 17	Rack 18	Rack 19	Rack 20
	alinhamento 3	Rack 21	Rack 22	Rack 23	Rack 24	Rack 25	Rack 26	Rack 27	Rack 28	Rack 29	Rack 30
		1	2	3	4	5	6	7	8	9	10
		1	2	3	4	5	6	7	8	9	10
	alinhamento 4	Rack 31	Rack 32	Rack 33	Rack 34	Rack 35	Rack 36	Rack 37	Rack 38	Rack 39	Rack 40
	alinhamento 5	Rack 41	Rack 42	Rack 43	Rack 44	Rack 45	Rack 46	Rack 47	Rack 48	Rack 49	Rack 50
		2	3	4	5	6	7	8	9	10	11
		2	3	4	5	6	7	8	9	10	11
	alinhamento 6	Rack 51	Rack 52	Rack 53	Rack 54	Rack 55	Rack 56	Rack 57	Rack 58	Rack 59	Rack 60
	alinhamento 7	Rack 61	Rack 62	Rack 63	Rack 64	Rack 65	Rack 66	Rack 67	Rack 68	Rack 69	Rack 70
		3	4	5	6	7	8	9	10	11	12
		3	4	5	6	7	8	9	10	11	12
	alinhamento 8	Rack 71	Rack 72	Rack 73	Rack 74	Rack 75	Rack 76	Rack 77	Rack 78	Rack 79	Rack 80
	alinhamento 9	Rack 81	Rack 82	Rack 83	Rack 84	Rack 85	Rack 86	Rack 87	Rack 88	Rack 89	Rack 90
		4	5	6	7	8	9	10	11	12	13
		4	5	6	7	8	9	10	11	12	13
	alinhamento 10	Rack 91	Rack 92	Rack 93	Rack 94	Rack 95	Rack 96	Rack 97	Rack 98	Rack 99	Rack 100
				Rack 54							
			5	Nível 5							
			4	Nível 4							
			3	Nível 3							
			2	Nível 2							
			1	Nível 1							
				5							

Figura A.7: layout 8 - instância de 500 paletes

## Apêndice B

### Outros anexos

Juntamente com este documento, em formato digital, colocamos outros anexos que seriam impossíveis colocar neste documento. Em formato digital encontra-se todos os layouts usados neste projeto, os ficheiros '.mod' e '.dat' usados no gusek em cada uma das três fases e para cada instância de paletes simulada. Encontra-se o ficheiro 'run' usado no 'neoServer' para obter os resultados da nossa simulação, bem como todos os resultados obtidos do neoServer para cada instância de paletes em cada uma das três fases.