



# Armazém de Paletes



## **Investigação Operacional** ENGENHARIA DE TELECOMUNICAÇÕES E INFORMÁTICA **2019/2020**

### **Docente:**

-José António Vasconcelos Oliveira

### **Trabalho realizado por:**

-Afonso Miguel Moreira de Queirós Pacheco, **A89397**

-Francisco Duarte Gomes Neto, **A90297**

-Rui Filipe Ribeiro Freitas, **A84121**

-Sandro Teixeira Ribeiro, **A85316**

-Tiago João Pereira Ferreira, **A85392**



# Índice

<b>Introdução .....</b>	<b>4</b>
<b>Layouts .....</b>	<b>5</b>
<b>Fase I - Problema de Afetação .....</b>	<b>7</b>
<b>Fase II - Problema de Afetação Generalizado .....</b>	<b>9</b>
<b>Fase III – Problema de Afetação Generalizado com restrição min-máx da altura do alvéolo .....</b>	<b>10</b>
<b>Fase IV - Estudo Multi-objetivo do Problema de Afetação Generalizado com restrição min-max da altura do alvéolo .....</b>	<b>12</b>
<b>Fase V – Estudo Multi-objetivo do Problema de Afetação Generalizado com restrição min-máx da altura do alvéolo. ....</b>	<b>15</b>
<b>Conclusão .....</b>	<b>16</b>
<b>Bibliografia .....</b>	<b>17</b>

# Introdução

O projeto “Armazém de Paletes” surge no âmbito da UC de Investigação Operacional com o objetivo de otimizar a gestão de um dado armazém. Este armazém contém 2 tipos de paletes: industriais e europeias que têm por objetivo o armazenamento de produtos. Estas paletes são guardadas em níveis de “*Racks*” espalhadas pelo armazém de acordo com *layouts* diferentes de cada elemento do grupo.

A partir de cada *layout* a gestão do armazém é feita seguindo as 5 etapas do projeto que passa por encontrar qual o melhor *layout* de todos os membros tendo em conta a distância percorrida pelos operadores e assim minimizar o custo associado à movimentação de paletes.

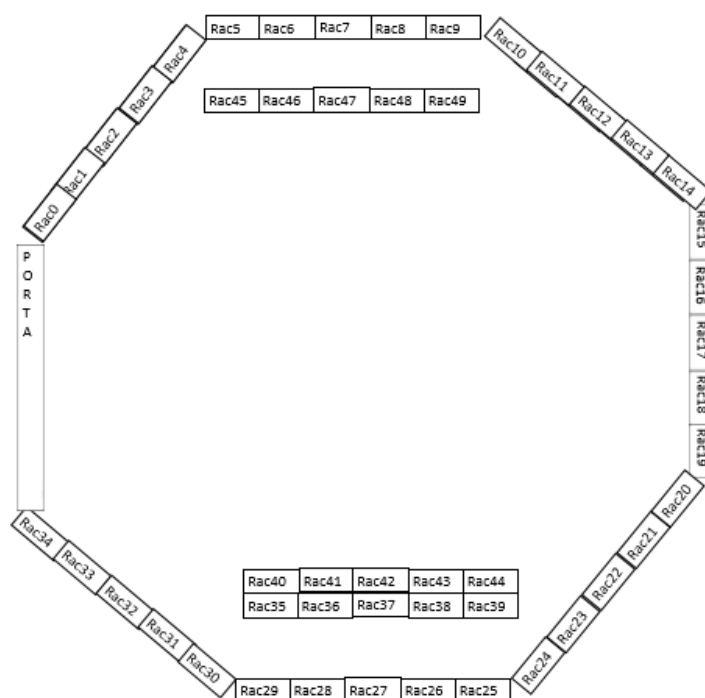
Este projeto está dividido em 5 etapas, a fase 1 passa por minimizar a distância percorrida pelo operador tendo em conta que em cada nível de *Rack* só cabe 1 paleta. Na fase 2 já é possível colocar em cada nível mais do que uma paleta respeitando a largura disponível em cada *Rack*. Quanto à fase 3, é adicionada uma altura máxima de nível em que esta é dada pela altura da paleta maior presente nesse nível. A fase 4 passa por estudar o desperdício linear(vertical) das paletes num dado nível, isto é, o espaço que sobra na afetação de paletes com diferenças significativas. Na 5ª e última fase deverá ser desenvolvida uma metodologia heurística implementada numa linguagem de programação para a afetação de paletes aos níveis minimizando ao máximo o tempo de movimentação assim como o desperdício vertical.

Como o grupo é constituído por 5 elementos foram criados 5 layouts diferentes cada um com 50 *Racks* com 10 níveis cada. Ao longo da resolução deste trabalho vamos chegar à informação de qual é o melhor e mais eficiente layout. Em relação às instâncias foram utilizadas 6 paletes para a micro, 60 para a mini, 300, 350, 400, 450 e 500 para as médias e 2000 para a grande.

# Layouts

Porta	
A1	A2
RACK 0	RACK 25
RACK 1	RACK 26
RACK 2	RACK 27
RACK 3	RACK 28
RACK 4	RACK 29
RACK 5	RACK 30
RACK 6	RACK 31
RACK 7	RACK 32
RACK 8	RACK 33
RACK 9	RACK 34
RACK 10	RACK 35
RACK 11	RACK 36
RACK 12	RACK 37
RACK 13	RACK 38
RACK 14	RACK 39
RACK 15	RACK 40
RACK 16	RACK 41
RACK 17	RACK 42
RACK 18	RACK 43
RACK 19	RACK 44
RACK 20	RACK 45
RACK 21	RACK 46
RACK 22	RACK 47
RACK 23	RACK 48
RACK 24	RACK 49

**FIGURA 1-** LAYOUT 1, SANDRO RIBEIRO



**FIGURA 2-** LAYOUT 2, FRANCISCO NETO

Rack 1	Rack 2	Rack 3	Rack 4	Rack 5	Porta	Rack 6	Rack 7	Rack 8	Rack 9	Rack 10
Rack 11	Rack 12	Rack 13	Rack 14	Rack 15		Rack 16	Rack 17	Rack 18	Rack 19	Rack 20
Rack 21	Rack 22	Rack 23	Rack 24	Rack 25		Rack 26	Rack 27	Rack 28	Rack 29	Rack 30
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		Rack 46	Rack 47	Rack 48	Rack 49	Rack 50

**FIGURA 3-** LAYOUT 3, TIAGO FERREIRA

PORTA	Alinhamento 1	RACK 0	RACK 1	RACK 2	RACK 3	RACK 4	RACK 5	RACK 6	RACK 7	RACK 8	RACK 9	RACK 10	RACK 11		
														RACK 48	
	Alinhamento 2	RACK 12	RACK 13	RACK 14	RACK 15	RACK 16	RACK 17	RACK 18	RACK 19	RACK 20	RACK 21	RACK 22	RACK 23		
	Alinhamento 3	RACK 24	RACK 25	RACK 26	RACK 27	RACK 28	RACK 29	RACK 30	RACK 31	RACK 32	RACK 33	RACK 34	RACK 35		
														RACK 49	
	Alinhamento 4	RACK 36	RACK 37	RACK 38	RACK 39	RACK 40	RACK 41	RACK 42	RACK 43	RACK 44	RACK 45	RACK 46	RACK 47		

**FIGURA 4-** LAYOUT 4, RUI FREITAS

Alinhamento 1			rak 0	Porta	rak 1	
	Corredor 1					
Alinhamento 2	rak 2	rak 3	rak 4		rak 5	rak 6
Alinhamento 3	rak 8	rak 9	rak 10		rak 11	rak 12
	Corredor 2					
Alinhamento 4	rak 14	rak 15	rak 16		rak 17	rak 18
Alinhamento 5	rak 20	rak 21	rak 22		rak 23	rak 24
	Corredor 3					
Alinhamento 6	rak 26	rak 27	rak 28		rak 29	rak 30
Alinhamento 7	rak 32	rak 33	rak 34		rak 35	rak 36
	Corredor 4					
Alinhamento 8	rak 38	rak 39	rak 40		rak 41	rak 42
Alinhamento 9	rak 44	rak 45	rak 46		rak 47	rak 48
	Corredor 5					

**FIGURA 5-** LAYOUT 5, AFONSO PACHECO

## Fase I - Problema de Afetação

Neste problema pretende-se que seja 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 nível (alvéolo / prateleira) só pode armazenar uma paleta.

O objetivo é minimizar o tempo total de acesso a todas as paletes, para isso definimos uma função objetivo chamada “TempoTotal” onde esta soma os tempos todos de acesso as diferentes paletes. Definimos uma restrição para que cada nível só possa conter uma paleta e outra restrição para que cada *Rack* apenas utilize o nível 1.

Para obtenção de resultados usamos o NEOS Server onde para cada instância submetíamos o ficheiro que contia os tempos “fase1.dat”, o ficheiro que contia o código com as restrições necessárias “fase1.mod” e o ficheiro que fornecia as instruções ao NEOS Server.

### CÓDIGO:

```
1  param N_paletes;  
2  
3  param N_Niveis;  
4  
5  param N_Racks;  
6  
7  set P:={i in 1..N_paletes};  
8  
9  set N:={j in 1..N_Niveis};  
10  
11 set R:={k in 1..N_Racks};  
12  
13 param Paleta{i in P,k in 1..4};  
14  
15 param Nivel{j in N,k in 1..3};  
16  
17 var x{i in P, j in N}, binary;  
18  
19 minimize TempoTotal: sum{i in P, j in N} Paleta[i,4]*Nivel[j,1]*x[i,j];  
20  
21 subject to Afeta_Paletes{i in P}:sum{j in N}x[i,j]=1;  
22  
23 s.t. Afeta_Niveis{j in N}:sum{i in P}x[i,j]<=1;
```

- **Resultados:**

		Micro	Mini	Média 1	Média 2	Média 3	Média 4	Média 5	Grande
<b>FASE 1</b>		<b>6</b>	<b>60</b>	<b>300</b>	<b>350</b>	<b>400</b>	<b>450</b>	<b>500</b>	<b>2000</b>
<b>Sandro</b>	<b>Layout 1</b>	32	2135	91688	119711	146474	176980	221287	
<b>Francisco</b>	<b>Layout 2</b>	32	2046	71480	91137	109696	130866	161754	
<b>Tiago</b>	<b>Layout 3</b>	84	1910	63037	80855	98076	117713	146291	
<b>Rui</b>	<b>Layout 4</b>	32	1791	68810	89042	108355	130384	162411	
<b>Afonso</b>	<b>Layout 5</b>	14	1419	50607	65447	80016	96783	121169	

**QUADRO 1-** SOLUÇÕES ÓTIMAS DA FASE 1

Com estes resultados obtidos podemos concluir que o Layout 5 para já é o melhor Layout e o Layout 1 o pior, desta forma podemos verificar que o layout 5 desempenhará um melhor papel para a solução pretendida nesta fase do problema. Sendo assim, o investidor deveria ter em atenção os dados fornecidos pelos solvers, fazendo assim um investimento ponderado e estudado para que sejam assegurados desperdícios de tempo e de tamanho desnecessário dentro do armazém.



## Fase II - Problema de Afetação Generalizado

Na Fase II pretende-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*.

Nesta fase, comparativamente à anterior, podemos ter mais do que uma paleta presente em cada nível, adicionando ao código anterior a restrição de que a largura das paletes num nível tem de ser menor ou igual à largura do nível.

### CÓDIGO:

```
param N_paletes;

param N_Niveis;

param N_Racks;

set P:={i in 1..N_paletes};

set N:={j in 1..N_Niveis};

set R:={k in 1..N_Racks};

param Paleta{i in P,k in 1..4};

param Nivel{j in N,k in 1..3};

var x{i in P, j in N}, binary;

minimize TempoTotal: sum{i in P, j in N} Paleta[i,4]*Nivel[j,1]*x[i,j];

subject to Afeta_Paletes{i in P}: sum{j in N} x[i,j]=1;

s.t. Afeta_Niveis{j in N}: sum{i in P} Paleta[i,3]*x[i,j]<=Nivel[j,2];
```

### • Resultados:

		Micro	Mini	Média 1	Média 2	Média 3	Média 4	Média 5	Grande
FASE 2		6	60	300	350	400	450	500	2000
Sandro	Layout 1	0	708	44073	57786	70798	85531	106695	
Francisco	Layout 2	0	708	40845	52543	63244	75165	92130	
Tiago	Layout 3	70	1026	36917	46845	56035	66200	80766	
Rui	Layout 4	0	675	36055	46725	56685	67839	83841	
Afonso	Layout 5	0	471	28507	36504	43765	51960	63840	

QUADRO 2- SOLUÇÕES ÓTIMAS DA FASE 2

## Fase III – Problema de Afetação Generalizado com restrição min-máx da altura do alvéolo

Na Fase III pretende-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 altura do nível (alvéolo / prateleira), dimensão  $A$ , será determinada em função das paletes que são lá colocadas, isto é, a paleta com maior altura do conjunto de paletes afetas ao nível definirá a altura do nível.

Para definir a altura máxima que uma *Rack* pode ter tivemos que escolher um empilhador que seria utilizado no armazém e que definia a altura máxima. Optamos por escolher um empilhador que permitia uma elevação de 9.29 metros.



<https://www.empilhoelectrico.pt/order-picker-tri-laterais-bt-vce-125-asf>

Acrescentamos as restrições necessárias. Uma restrição para que a altura de cada nível seja determinada pela paleta com maior altura e uma restrição para definir a altura máxima da *Rack*.

### CÓDIGO:

```

param H_MAX_RACK;

param N_paletes;

param N_Niveis;

param N_Racks;

set P:={i in 1..N_paletes};

set N:={j in 1..N_Niveis};

set R:={k in 1..N_Racks};

param Palette{i in P,k in 1..4};

param Nivel{j in N,k in 1..3};

var x{i in P, j in N}, binary;

var altura_nivel{j in N}, >=0;

var altura_rack{k in R}, >=0;

minimize TempoTotal: sum{i in P, j in N} Palette[i,4]*Nivel[j,1]*x[i,j];

subject to Afeta_Paletes{i in P}: sum{j in N} x[i,j]=1;

subject to Afeta_Niveis{j in N}: sum{i in P} x[i,j]*Palette[i,3]<=Nivel[j,2];

subject to def_altura_nivel{i in P, j in N}: altura_nivel[j]>=Palette[i,2]*x[i,j];

subject to def_altura_Rack{k in R}: altura_rack[k]=sum{j in N: Nivel[j,3]=k} altura_nivel[j];

subject to def_altura_Rack2{k in R}: altura_rack[k]<=H_MAX_RACK;

```

### • Resultados:

FASE 3		Micro	Mini	Média 1	Média 2	Média 3	Média 4	Média 5	Grande
		6	60	300	350	400	450	500	2000
Sandro	Layout 1	0	708	44073	57778	70797	85536	106692	
Francisco	Layout 2	0	708	40846	52540	63241	75158	92130	
Tiago	Layout 3	70	1026	36917	46842	56038	66197	80766	
Rui	Layout 4	0	675	36053	46725	56685	67839	83839	
Afonso	Layout 5	0	471	28506	36503	43765	51962	63840	

**QUADRO 3-** SOLUÇÕES ÓTIMAS DA FASE 3

## Fase IV - Estudo Multi-objetivo do Problema de Afetação Generalizado com restrição min-max da altura do alvéolo

Na Fase IV pretende-se continuar o estudo da fase III mais o desperdício(vertical).

Como nesta fase se pretende minimizar o desperdício tivemos que adicionar as restrições para tal.

Para esta fase foi apenas utilizada a instância mini (60 paletes) e o melhor layout da fase 3. O melhor layout foi o layout 5 que obteve melhores resultados durante as várias experiências realizadas.

Ao minimizar o desperdício vertical verificou-se que o layout apresentava um desperdício nulo, mas o tempo total para aceder as paletes elevado. Criou se então uma restrição para o tempo total e realizou-se várias experiências. Obteve-se o menor tempo total de 599 segundos mantendo o desperdício nulo. Com valores inferiores a 599 o desperdício começou a aumentar.

Ao minimizar o tempo total constatou-se que para a solução obtida na fase III (471s) o desperdício era elevado. Criou se então uma restrição para o desperdício e realizou-se novamente várias experiências. Obteve-se desperdício nulo quando o tempo foi equivalente a 523 segundos.

## CÓDIGO:

```
param H_MAX_RACK;

param N_paletes;

param N_Niveis;

param N_Racks;

set P:={i in 1..N_paletes};

set N:={i in 1..N_Niveis};

set R:={j in 1..N_Racks};

param Palette(i in P, 1..4);

param Nivel(j in N, 1..3);

var x(i in P, j in N), binary;

var altura_nivel(j in N), >=0;

var altura_rack(k in R), >=0;

var desp_vert(i in P), >=0;

var aux_fol, >=0;

var aux_fo2, >=0;

#minimize TempoTotal: sum(i in P, j in N) Palette[i,4]*Nivel[j,1]*x[i,j];

minimize DespTotal: sum(i in P) desp_vert[i];

subject to Afeta_Paletes(i in P): sum(j in N) x[i,j]=1;

subject to Afeta_Niveis(j in N): sum(i in P) x[i,j]*Palette[i,3]<=Nivel[j,2];

subject to def_altura_nivel(i in P, j in N): altura_nivel[j]>=Palette[i,2]*x[i,j];

subject to def_altura_Rack(k in R): altura_rack[k]=sum(j in N: Nivel[j,3]=k) altura_nivel[j];

subject to def_altura_Rack2(k in R): altura_rack[k]<=H_MAX_RACK;

subject to def_desp_vert(i in P, j in N): x[i,j]=1 ==> (altura_nivel[j]-Palette[i,2]*x[i,j]) = desp_vert[i];

subject to def_aux_fol: aux_fol = sum(i in P, j in N) Palette[i,4]*Nivel[j,1]*x[i,j];

subject to def_aux_fo2: aux_fo2 = sum(i in P) desp_vert[i];

subject to desp_max: aux_fol <= 480;

#subject to desp_max: aux_fo2 <= 20;
```

- Resultados:

FASE 4	Micro	Mini	Média 1	Média 2	Média 3	Média 4	Média 5	Grande	
	6	60	300	350	400	450	500	2000	
	Mini	min FO2	Layout 5			Mini	min FO1	Layout 5	
		s.t. FO1 <=	FO1	FO2			s.t. FO2 <=	FO1	FO2
	Sandro	9999999	7037	0		Sandro	9999999	471	516
	Francisco	7000	5464	0		Francisco	300	471	296
	Tiago	5000	4994	0		Tiago	200	472	184
	Rui	3000	2830	0		Rui	150	472	144
	Afonso	2000	1990	0		Afonso	120	475	116
	Sandro	1000	989	0		Sandro	90	476	90
	Francisco	900	891	0		Francisco	80	476	80
	Tiago	800	778	0		Tiago	70	479	70
	Rui	700	699	0		Rui	60	484	54
	Afonso	600	599	0		Afonso	50	485	44
	Sandro	500	499	14		Sandro	40	485	40
	Francisco	490	490	36		Francisco	30	488	24
	Tiago	480	479	64		Tiago	20	496	14
	Rui	475	475	116		Rui	10	501	4
	Afonso	471	471	204		Afonso	0	523	0

## Fase V – Estudo Multi-objetivo do Problema de Afetação Generalizado com restrição min-máx da altura do alvéolo.

Na fase V foi proposto o desenvolvimento de uma metodologia heurística implementada numa linguagem à nossa escolha em que optamos por utilizar Java pois foi com a qual mais trabalhamos este ano, tanto em Paradigmas da Programação I como II.

Nesta fase de desenvolvimento do projeto, tivemos de realizar uma revisão a todas restrições até á Fase III, já devidamente apontadas anteriormente, tendo como principal intuito acautelar o tempo total de extração de paletes e com isto reduzi-lo.

Conseguimos concluir o que nos foi pedido com sucesso onde neste código conseguimos obter a informação do tempo de movimentação, juntamente com o desperdício (vertical) como apresentado nos anexos seguintes:

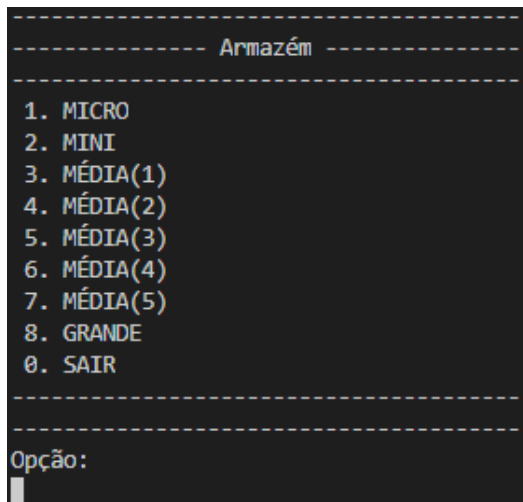


FIGURA 7- MENU

```

Altura:79.0
Paletes:[256, 256, 253]

Rack:46
Nível:10
Altura:59.0
Paletes:[252, 253, 253]

Rack:50
Nível:9
Altura:79.0
Paletes:[254, 254, 251]

Rack:35
Nível:10
Altura:79.0
Paletes:[253, 251, 249]

Rack:47
Nível:10
Altura:79.0
Paletes:[256, 251, 251]

Rack:49
Nível:10
Altura:79.0
Paletes:[251]

Tempo Total:427141
Desperdicio Total:18294.6
  
```

FIGURA 8- SOLUÇÃO ÓTIMA ENCONTRADA PARA INSTÂNCIA

## Conclusão

Ao longo da realização deste trabalho, o grupo enfrentou várias dificuldades, que com o espírito crítico e trabalho de equipa se superaram. Trabalho este que serviu para aplicar conceitos adquiridos na UC a um problema real tornando assim a sua resolução mais interessante. O Gusek e o Neos server foram os solvers utilizados na resolução deste trabalho.

Com a conclusão deste projeto prático, verificamos que nossa capacidade de este tipo de problemas terá sido reforçada. Assim sendo, num futuro que se encontra cada vez mais próximo o nosso grupo estará preparado para encarar, resolver e solucionar problemas de afetação bastante importantes para aqueles que deles dependem.

Em suma, este trabalho para além por á prova os conceitos lecionados na UC, foi também um desafio na exploração dos solvers e que contribuiu para novas experiências fundamentais no desenvolvimento de faculdades que serão essências para a modelação neste e deste tipo de problemas cada vez mais frequentes na realidade dos dias hoje.



## Bibliografia

Oliveira, J. A. (2019 / 2020) - Armazem\_Paletes\_v21.

Oliveira, J. A. (2019 / 2020) - Estudo Multi-objetivo, 2020, Universidade do Minho, Guimarães.

A.J.M.Guimarães Rodrigues, Investigação Operacional - Modelos Determinísticos, Vol.I, Universidade do Minho, 1993.