Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Investigação Operacional

Ano Letivo de 2024/2025

Trabalho Prático 1

João Delgado	**Tomás Machado**	**Hugo Rauber**	**Simão Mendes**	**Nelson Rocha**
A106836	A104186	A104534	A106928	A106884

Março, 2025

Índice

Lista de Figuras

Lista de Tabelas

1. Introdução

1.1. Descrição

Este projeto é uma componente fundamental da disciplina de Investigação Operacional do ano académico 2024/2025. A unidade curricular tem como objetivo principal introduzir aos alunos os conceitos essenciais da otimização, proporcionando uma base sólida que permita compreender e aplicar técnicas de otimização em diversos contextos. Além disso, a disciplina promove o desenvolvimento de habilidades analíticas, que são cruciais para a resolução eficaz de problemas complexos que podem surgir em situações do mundo real.

1.2. Motivação:

Através deste projeto, o nosso grupo tem como meta aprofundar o conhecimento na resolução de problemas práticos, com especial ênfase no problema de empacotamento. Este tipo de problema é relevante em várias áreas, como logística, gestão de armazéns e transporte, onde a eficiência no uso do espaço e a minimização de custos são essenciais. Além disso, uma das nossas principais motivações é garantir a aprovação na disciplina, não apenas cumprindo os requisitos, mas também alcançando um desempenho de qualidade que reflita o nosso esforço e dedicação.

2. Desenvolvimento

2.0. Questão Zero

No âmbito deste projeto, o valor de **xABCDE** foi determinado como **106928**. As partições deste número são definidas da seguinte forma: **x=1**, **A=0**, **B=6**, **C=9**, **D=2** e **E=8**. Com base nestes valores, elaboramos duas tabelas que ilustram os resultados obtidos.

A primeira tabela apresenta o número de contentores utilizados e o respetivo comprimento de cada um, enquanto a segunda tabela detalha o número de itens a serem empacotados e o comprimento correspondente de cada item. As tabelas e os cálculos resultantes são apresentados de seguida:

Quantidade	**Comprimento**
Ilimitada	11
7 (B + 1)	10
3 (D + 1)	7

Tabela 1: Contentores disponíveis.

Quantidade	**Comprimento**
0 (B é par, logo k1 = 0)	1
17 (C é impar, logo k2 = C + 2)	2
0 (D é par, logo k3 = 0)	3
10 (E é par, logo k4 = E + 2)	4
5	5

Tabela 2: Itens a empacotar.

A **soma total dos comprimentos** dos itens a empacotar é $0 \cdot 1 + 17 \cdot 2 + 0 \cdot 3 + 10 \cdot 4 + 5 \cdot 5 = 99$, que obviamente se encontra dentro do limite dos contentores que é ilimitada.

2.1. Questão Um

Neste problema, o objetivo é **minimizar** a quantidade de contentores utilizados para empacotar um conjunto de itens, de forma a reduzir a soma dos comprimentos dos contentores empregados. Para tal, é necessário determinar a quantidade de contentores de cada comprimento a utilizar, respeitando a disponibilidade de contentores existentes e a demanda dos itens.

Definem-se os seguintes conceitos:

Parâmetros:

- $W_k =$ Diferentes capacidades de cada classe de contentor.
- $B_k =$ Número de Contentores disponível em cada classe.
- $w_i =$ Diferentes tamanhos de cada tipo de item.
- $b_i =$ Número de itens para empacotar dentro de cada tipo.

Variáveis:

- $z_k =$ Número de contentores de capacidade W_k utilizados.
- $x_{\{d,e\}}$ Número de itens de tamanho $e - d$ colocado em um contentor a d unidades do seu início.

Domínios:

- k representa as classes de diferentes contentores.
- i representa os diferentes tipos de itens disponíveis.
- A representa todos os arcos possíveis que cumprem as condições $A = \{(d, e) : 0 \leq d < e \leq W_{max} \wedge e - d = w_i \forall 1 \leq i \leq m\}$.
- A' é um subconjunto de A proveniente da utilização de critérios de redução sobre o mesmo. É especificado na subsecção de **Critérios de Redução**.

Deste modo, aplicando a informação discutida anteriormente ao nosso problema obtivemos o seguinte:
- $W_k = \{11, 10, 7\}$, com $W_{max} = W_1 = 11$, $W_2 = 10$ e $W_3 = 7$.
- $B_k = \{\infty, 7, 3\}$, com $B_1 = \infty$, $B_2 = 7$ e $B_3 = 3$.
- $w_i = \{5, 4, 2\}$, com $w_1 = 5$, $w_2 = 4$ e $w_3 = 2$.
- $b_i = \{5, 10, 17\}$, com $b_1 = 5$, $b_2 = 10$, $b_3 = 17$.

Tendo em conta o tamanho dos itens podemos concluir que o conjunto A é:
- $A = \{(0, 2), (0, 4), (0, 5), (1, 3), (1, 5), (1, 6), (2, 4), (2, 6), (2, 7), (3, 5), (3, 7), (3, 8), (4, 6), (4, 8)$ $(4, 9), (5, 7), (5, 9), (5, 10), (6, 8), (6, 10), (6, 11), (7, 9), (7, 11), (8, 10), (9, 11)\}$

É importante mencionar que para um bom funcionamento do modelo tanto os itens como os contentores foram ordenados de forma decrescente pela sua capacidade.

Modelo utilizado:

Para formular o modelo de programação linear (LP) utilizado no decorrer deste trabalho, utilizou-se o **modelo de fluxos em arcos**, designado na literatura anglo-saxônica como *arc-flow model*.
É importante referir que toda a modelação matemática é proveniente dos trabalhos apresentados na **Bibliografia**.

Variáveis de Decisão:

Considere-se as variáveis de decisão x_{de} associada aos arcos dos itens, que correspondem ao número de itens de tamanho $e - d$ inseridos no contentor a distância de d unidades do início do mesmo. Também se deve considerar a variável de decisão $z_k, k = 1, ..., K$, associada aos arcos que corresponde à quantidade de contentores de capacidade W_k utilizados. Também pode ser vista como um arco do vértice W_k ao vértice 0, tendo a possibilidade de ser denotada por $x_{\{W_k,0\}}$.

Objetivo:

Minimizar a soma dos comprimentos dos contentores utilizados, garantindo uma escolha eficiente dos contentores a serem utilizados.

Critérios de Redução:

Existem várias soluções alternativas com os mesmos itens em cada contentor. Para reduzir a simetria do espaço de soluções e o tamanho do modelo, podemos considerar apenas um subconjunto de arcos de $A' \in A$. Ao procurar uma solução com itens ordenados em tamanhos decrescentes, os critérios apresentados de seguida podem ser utilizados para limitar o número de arcos considerados.

1. Sejam w_{i_1} e w_{i_2} o tamanho de dois itens tal que $w_{i_1} > w_{i_2}$. O arco w_{i_2}, designado pelos pontos $\left(d, d + w_{i_2}\right)$ pode apenas ter a sua cauda no ponto d que é cabeça de outro arco de tamanho w_{i_1}, $\left(d - w_{i_1}\right)$, ou, num outro caso, no início do contentor. De forma simples este critério aponta que se o contentor tiver alguma perda do espaço de utilização, esta apenas ocorrerá no final do mesmo.
2. Todas as variáveis de arco que provoquem $x_{d,d+1}$ podem ser definidas como zero para $d < w_m$. Num contentor, o número de arcos consecutivos correspondentes a um único tamanho de item deve ser menor ou igual ao número de itens desse tamanho.
3. Sejam i_1 e i_2 dois tamanhos de itens, tais que $w_{i_1} > w_{i_2}$. Dado qualquer nó d que seja a cabeça de outro arco de tamanho w_{i_1} ou tal que $d = 0$, os únicos arcos válidos para o tamanho w_{i_2} são aqueles que partem dos nós $d + s \cdot w_{i_2}, s = 0, 1, ..., b_{i_2} - 1$, com a condição de que:

   $$d + s \cdot w_{i_2} \leq W_{max},$$

   onde b_{i_2} representa a procura (demanda) dos itens de tamanho w_{i_2}.

   Com estes critérios de redução podemos obter o subconjunto de arcos:

   $A' = \{(0, 2), (0, 4), (0, 5), (2, 4), (2, 6), (2, 7), (4, 6), (4, 8), (4, 9), (5, 7), (5, 9), (5, 10), (6, 8),$
   $(6, 10), (6, 11), (7, 9), (7, 11), (8, 10), (9, 11)\}$

Função Objetivo:

Este problema é formulado de modo a minimizar a soma das capacidades dos contentores que são necessários para empacotar todos os itens.

Minimize $\sum_{k=1}^{K} W_k \cdot z_k$

Tendo em conta que: $W_k = \{11, 10, 7\}$, com $W_1 = 11$, $W_2 = 10$ e $W_3 = 7$.

Esta função objetivo pode ser expandida ao caso específico do problema ficando como:

Minimize z = $11z_1 + 10z_2 + 7z_3$

Restrições:

Como definido no artigo *LP models for bin packing and cutting stock problems* [1], as restrições para este modelo podem ser vistas da seguinte forma:

$$-\sum_{(d,e)\in A'} x_{de} + \sum_{(e,f)\in A'} x_{ef} = \begin{cases} \sum_{k+1}^{K} z_k & \text{if } e = 0, \\ -z_k & \text{for } e = W_k \ , \quad k = 1,...,K, \\ 0 & \text{otherwise,} \end{cases}$$

$$\sum_{(d,d+w_i)\in A'} x_{d,d+w_i} \geq b_i, i = 1, ..., m,$$

$$z_k \leq B_k, k = 1, ..., K,$$

$$x_{de} \geq 0 \text{ and integer } \forall (d,e) \in A',$$

$$z_k \geq 0 \text{ and integer k = 1,...,K.}$$

A primeira restrição, utilizada para controlo de fluxo, pode ser expandida nos seguintes casos:

Quando o $e = 0$:

$-\sum_{(d,0)\in A'} x_{d0} + \sum_{(0,f)\in A'} x_{0f} = \sum_{k+1}^{K} z_k \Leftrightarrow x_{05} + x_{04} + x_{02} = z_1 + z_2 + z_3$

Quando $e = W_k$:

$-\sum_{(d,11)\in A'} x_{d11} + \sum_{(11,f)\in A'} x_{11f} = -z_1 \Leftrightarrow -x_{611} - x_{711} - x_{911} = -z_1$

$-\sum_{(d,10)\in A'} x_{d10} + \sum_{(10,f)\in A'} x_{10f} = -z_2 \Leftrightarrow -x_{510} - x_{610} - x_{810} = -z_2$

$-\sum_{(d,7)\in A'} x_{d7} + \sum_{(7,f)\in A'} x_{7f} = -z_3 \Leftrightarrow -x_{27} - x_{57} = -z_3$

Quando $e \in \{2, 4, 5, 6, 8, 9\}$:

$-\sum_{(d,2)\in A'} x_{d2} + \sum_{(2,f)\in A'} x_{2f} = 0 \Leftrightarrow -x_{02} + x_{24} + x_{26} + x_{27} = 0$

$-\sum_{(d,4)\in A'} x_{d4} + \sum_{(4,f)\in A'} x_{4f} = 0 \Leftrightarrow -x_{04} - x_{24} + x_{46} + x_{48} + x_{49} = 0$

$-\sum_{(d,5)\in A'} x_{d5} + \sum_{(5,f)\in A'} x_{5f} = 0 \Leftrightarrow -x_{05} + x_{57} + x_{59} + x_{510} = 0$

$-\sum_{(d,6)\in A'} x_{d6} + \sum_{(6,f)\in A'} x_{6f} = 0 \Leftrightarrow -x_{26} - x_{46} + x_{68} + x_{610} + x_{611} = 0$

$-\sum_{(d,8)\in A'} x_{d8} + \sum_{(8,f)\in A'} x_{8f} = 0 \Leftrightarrow -x_{48} - x_{68} + x_{810} = 0$

$-\sum_{(d,9)\in A'} x_{d9} + \sum_{(9,f)\in A'} x_{9f} = 0 \Leftrightarrow -x_{49} - x_{59} - x_{79} + x_{911} = 0$

A segunda restrição pode ser expandida para:

$\sum_{(d,d+2)\in A'} x_{d,d+2} \geq 17 \Leftrightarrow x_{02} + x_{24} + x_{46} + x_{57} + x_{68} + x_{79} + x_{810} + x_{911} \geq 17$

$\sum_{(d,d+4)\in A'} x_{d,d+4} \geq 10 \Leftrightarrow x_{04} + x_{26} + x_{48} + x_{59} + x_{610} + x_{711} \geq 10$

$\sum_{(d,d+5)\in A'} x_{d,d+5} \geq 5 \Leftrightarrow x_{05} + x_{27} + x_{49} + x_{510} + x_{611} \geq 5$

A terceira restrição pode ser expandida como:

$0 \leq z_1 \leq \infty$

$0 \leq z_2 \leq 7$

$0 \leq z_3 \leq 3$

A quarta restrição será:

$x_{05}, x_{27}, x_{49}, x_{510}, x_{611}, x_{04}, x_{26}, x_{48}, x_{59}, x_{610}, x_{711}, x_{02}, x_{24}, x_{46}, x_{57}, x_{68}, x_{79}, x_{810}, x_{911} \geq 0$

$x_{05}, x_{27}, x_{49}, x_{510}, x_{611}, x_{04}, x_{26}, x_{48}, x_{59}, x_{610}, x_{711}, x_{02}, x_{24}, x_{46}, x_{57}, x_{68}, x_{79}, x_{810}, x_{911} \in \mathbb{Z}$

À semelhança da quarta, a quinta restrição fica:

$z_1, z_2, z_3 \geq 0$

$z_1, z_2, z_3 \in \mathbb{Z}$

2.2. Questão Dois

Modelo de Programação Linear via Fluxo de Arcos

No **modelo por fluxo de arcos**, cada contentor é representado como um grafo em que os nós correspondem aos níveis de preenchimento (de 0 até à capacidade máxima) e os arcos representam a inserção de itens com determinados comprimentos, de forma que a soma dos comprimentos não ultrapasse a capacidade do contentor. Cada arco possui uma variável de decisão que indica quantas vezes essa transição ocorre, e a formação de um caminho completo do nó 0 ao nó de capacidade máxima corresponde a um empacotamento válido, cujo custo é o associado à utilização do respetivo contentor. A função objetivo consiste em minimizar a quantidade dos contentores empregados, enquanto as restrições garantem a conservação do fluxo nos nós, a satisfação integral da demanda dos itens e o respeito à capacidade/disponibilidade dos contentores.

Função Objetivo:

Minimize $z = 11z_1 + 10z_2 + 7z_3$

Restrições:

- $x_{05} + x_{04} + x_{02} = z_1 + z_2 + z_3;$

- $-x_{611} - x_{711} - x_{911} = -z_1;$
- $-x_{510} - x_{610} - x_{810} = -z_2;$
- $-x_{27} - x_{57} = -z_3;$

- $-x_{02} + x_{24} + x_{26} + x_{27} = 0;$
- $-x_{04} - x_{24} + x_{46} + x_{48} + x_{49} = 0;$
- $-x_{05} + x_{57} + x_{59} + x_{510} = 0;$
- $-x_{26} - x_{46} + x_{68} + x_{610} + x_{611} = 0;$
- $-x_{48} - x_{68} + x_{810} = 0;$
- $-x_{49} - x_{59} - x_{79} + x_{911} = 0;$

- $x_{05} + x_{27} + x_{49} + x_{510} + x_{611} \geq 5;$
- $x_{04} + x_{26} + x_{48} + x_{59} + x_{610} + x_{711} \geq 10;$
- $x_{02} + x_{24} + x_{46} + x_{57} + x_{68} + x_{79} + x_{810} + x_{911} \geq 17;$

- $z_1 \geq 0;$
- $z_2 \leq 7;$
- $z_2 \geq 0;$
- $z_3 \leq 3;$
- $z_3 \geq 0;$

- $x_{05} \geq 0;$
- $x_{27} \geq 0;$
- $x_{49} \geq 0;$
- $x_{510} \geq 0;$
- $x_{611} \geq 0;$
- $x_{04} \geq 0;$

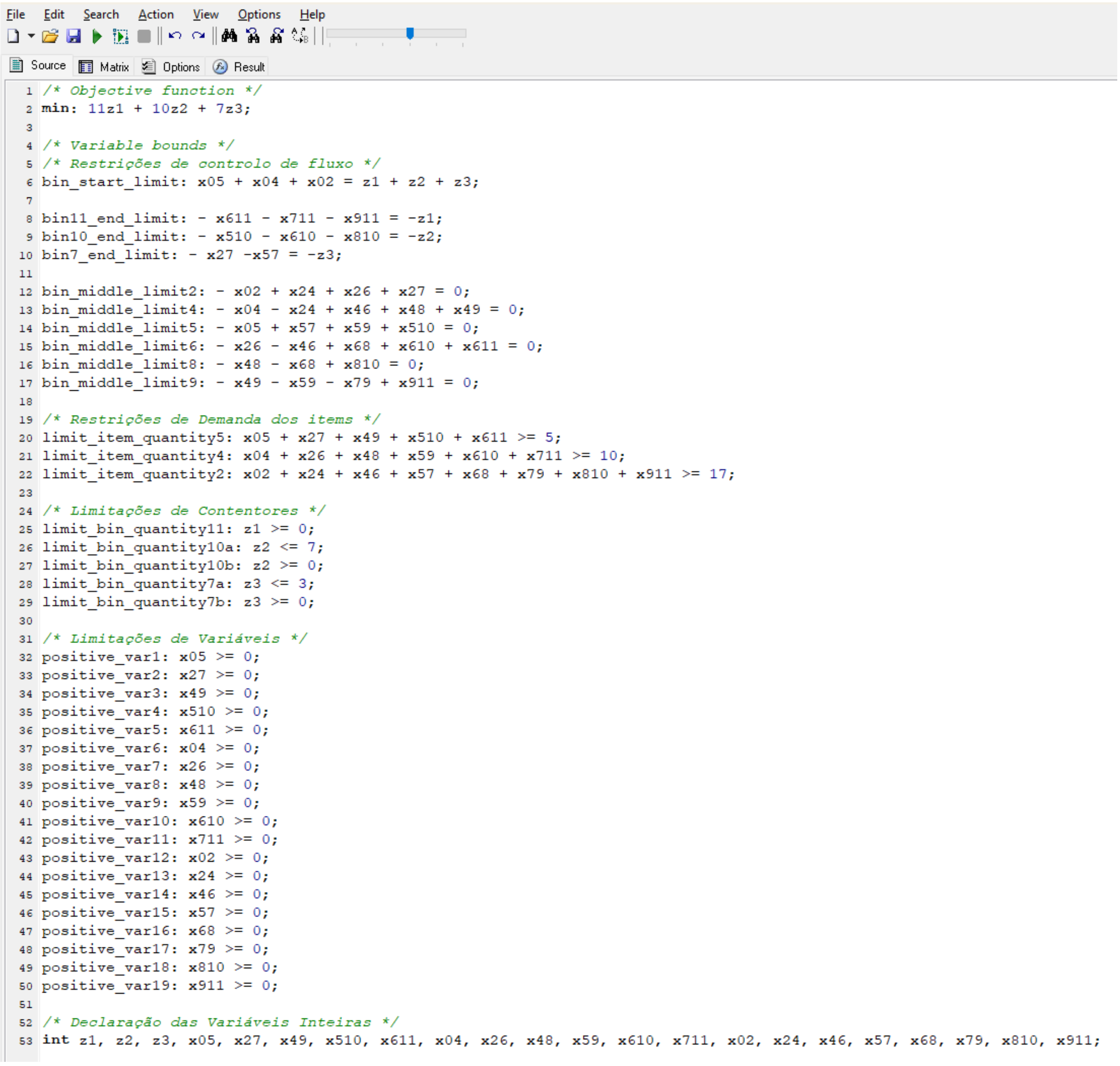
- $x_{26} \geq 0$;
- $x_{48} \geq 0$;
- $x_{59} \geq 0$;
- $x_{610} \geq 0$;
- $x_{711} \geq 0$;
- $x_{02} \geq 0$;
- $x_{24} \geq 0$;
- $x_{46} \geq 0$;
- $x_{57} \geq 0$;
- $x_{68} \geq 0$;
- $x_{79} \geq 0$;
- $x_{810} \geq 0$;
- $x_{911} \geq 0$;

- int $z_1, z_2, z_3, x_{05}, x_{27}, x_{49}, x_{510}, x_{611}, x_{04}, x_{26}, x_{48}, x_{59}, x_{610}, x_{711}, x_{02}, x_{24}, x_{46}, x_{57}, x_{68}, x_{79}, x_{810}, x_{911}$;

2.3. Questão Três

Apresentação do Ficheiro de Input

Nesta secção, apresenta-se o ficheiro de *input* utilizado para a implementação do modelo de empacotamento, extraído integralmente através da técnica de *cut-and-paste*. Este ficheiro foi estruturado de forma a conter todas as informações essenciais para a definição dos parâmetros, variáveis e restrições do problema de otimização, fundamentado no **modelo de fluxo de arcos**.



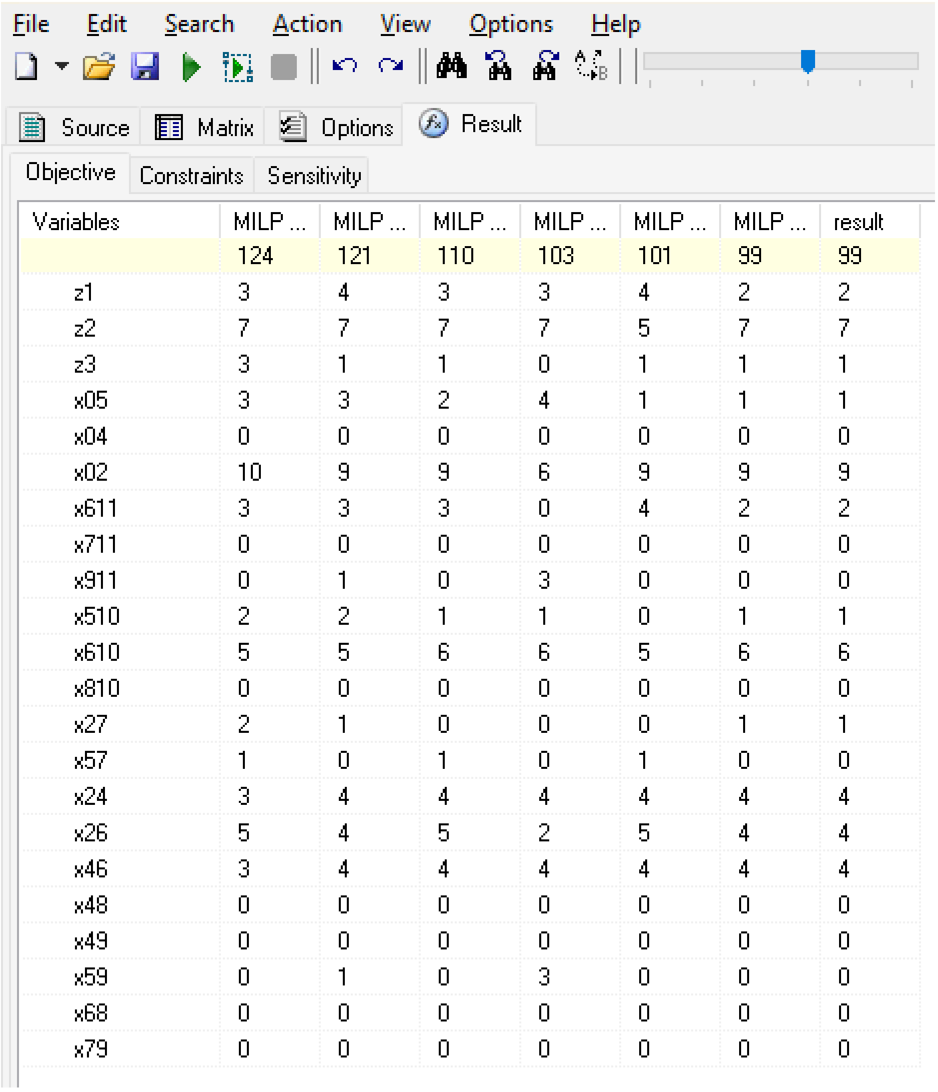

```
/* Objective function */
min: 11z1 + 10z2 + 7z3;

/* Variable bounds */
/* Restrições de controlo de fluxo */
bin_start_limit: x05 + x04 + x02 = z1 + z2 + z3;

bin11_end_limit: - x611 - x711 - x911 = -z1;
bin10_end_limit: - x510 - x610 - x810 = -z2;
bin7_end_limit: - x27 -x57 = -z3;

bin_middle_limit2: - x02 + x24 + x26 + x27 = 0;
bin_middle_limit4: - x04 - x24 + x46 + x48 + x49 = 0;
bin_middle_limit5: - x05 + x57 + x59 + x510 = 0;
bin_middle_limit6: - x26 - x46 + x68 + x610 + x611 = 0;
bin_middle_limit8: - x48 - x68 + x810 = 0;
bin_middle_limit9: - x49 - x59 - x79 + x911 = 0;

/* Restrições de Demanda dos items */
limit_item_quantity5: x05 + x27 + x49 + x510 + x611 >= 5;
limit_item_quantity4: x04 + x26 + x48 + x59 + x610 + x711 >= 10;
limit_item_quantity2: x02 + x24 + x46 + x57 + x68 + x79 + x810 + x911 >= 17;

/* Limitações de Contentores */
limit_bin_quantity11: z1 >= 0;
limit_bin_quantity10a: z2 <= 7;
limit_bin_quantity10b: z2 >= 0;
limit_bin_quantity7a: z3 <= 3;
limit_bin_quantity7b: z3 >= 0;

/* Limitações de Variáveis */
positive_var1: x05 >= 0;
positive_var2: x27 >= 0;
positive_var3: x49 >= 0;
positive_var4: x510 >= 0;
positive_var5: x611 >= 0;
positive_var6: x04 >= 0;
positive_var7: x26 >= 0;
positive_var8: x48 >= 0;
positive_var9: x59 >= 0;
positive_var10: x610 >= 0;
positive_var11: x711 >= 0;
positive_var12: x02 >= 0;
positive_var13: x24 >= 0;
positive_var14: x46 >= 0;
positive_var15: x57 >= 0;
positive_var16: x68 >= 0;
positive_var17: x79 >= 0;
positive_var18: x810 >= 0;
positive_var19: x911 >= 0;

/* Declaração das Variáveis Inteiras */
int z1, z2, z3, x05, x27, x49, x510, x611, x04, x26, x48, x59, x610, x711, x02, x24, x46, x57, x68, x79, x810, x911;
```

Figura 1: Captura de Ecrã do Ficheiro de Input do Modelo no LPSolve.

2.4. Questão Quatro

Apresentação do Ficheiro de Output

Nesta secção, é apresentado o ficheiro de *output* resultante da execução do programa, obtido através da técnica de *cut-and-paste*. O ficheiro de *output* contém toda a informação relevante que demonstra o desempenho e os resultados obtidos pelo modelo de empacotamento.



Variables	MILP ...	MILP ...	MILP ...	MILP ...	MILP ...	MILP ...	result
	124	121	110	103	101	99	99
z1	3	4	3	3	4	2	2
z2	7	7	7	7	5	7	7
z3	3	1	1	0	1	1	1
x05	3	3	2	4	1	1	1
x04	0	0	0	0	0	0	0
x02	10	9	9	6	9	9	9
x611	3	3	3	0	4	2	2
x711	0	0	0	0	0	0	0
x911	0	1	0	3	0	0	0
x510	2	2	1	1	0	1	1
x610	5	5	6	6	5	6	6
x810	0	0	0	0	0	0	0
x27	2	1	0	0	0	1	1
x57	1	0	1	0	1	0	0
x24	3	4	4	4	4	4	4
x26	5	4	5	2	5	4	4
x46	3	4	4	4	4	4	4
x48	0	0	0	0	0	0	0
x49	0	0	0	0	0	0	0
x59	0	1	0	3	0	0	0
x68	0	0	0	0	0	0	0
x79	0	0	0	0	0	0	0

Figura 2: Captura de Ecrã do Output da Execução do Modelo no LPSolve.

2.5. Questão Cinco

Apresentação da Solução

Esta subsecção do relatório tem como objetivo apresentar a solução ótima para este problema de empacotamento de itens em contentores, abordando três aspetos principais: o valor da solução ótima, a representação do grafo subjacente ao **modelo de fluxos em arcos** e o **plano de empacotamento** dos itens nos contentores utilizados.

(a) - Valor da Solução Ótima:

Será indicado o valor da solução ótima, que corresponde à soma dos comprimentos dos contentores utilizados para acomodar todos os itens. Este valor reflete a eficiência da solução encontrada, minimizando o espaço total ocupado. Assim sendo, como consta na **Secção 2.5**, a solução ótima é: **2** contentores de **tamanho 11**, **7** contentores de **tamanho 10** e **1** contentor de **tamanho 7**.

(b) - Grafo Subjacente ao Modelo de Fluxos em Arcos:

Será apresentado o grafo que serve de base ao modelo de fluxos em arcos, destacando os fluxos não-nulos associados à solução ótima. Esta representação gráfica permite visualizar como os itens são distribuídos entre os contentores, seguindo as restrições e objetivos do problema. Segue-se a imagem do grafo:

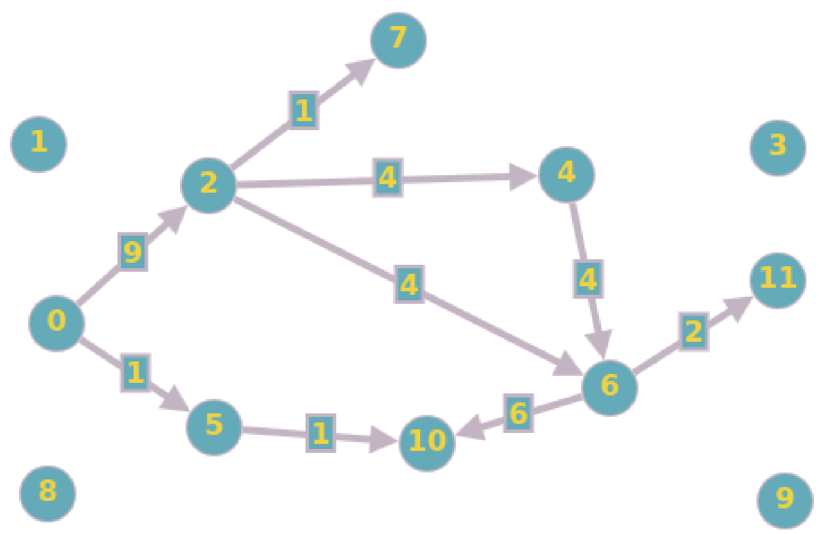


Figura 3: Grafo Subjacente ao Modelo de Fluxos em Arcos (Solução óptima).

(c) - Plano de Empacotamento:

Por fim, será exibido um possível plano de empacotamento ótimo, seguindo o *output* do *LPSolve*. No plano é possível identificar a distribuição ótima dos itens inseridos em cada um dos tipos de contentor. Segue-se a imagem do plano:

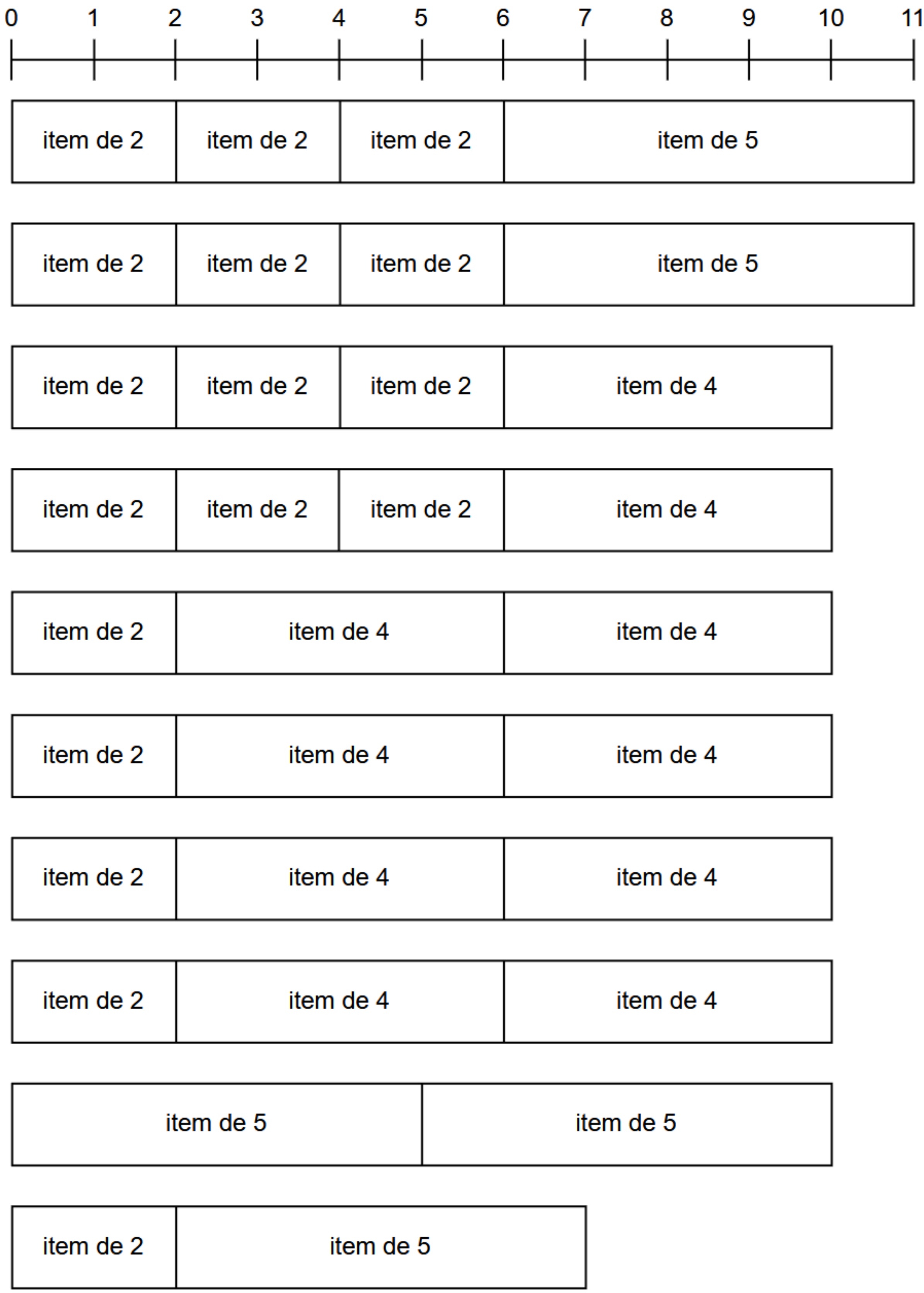


Figura 4: Possível plano de empacotamento ótimo.

2.6. Questão Seis

A validação do modelo é um passo fundamental para garantir que as soluções obtidas são corretas e coerentes com o problema proposto. Para tal, seguimos um conjunto de procedimentos que nos permitiram verificar a validade dos resultados obtidos.

2.6.1. Testes de Correção do Modelo

Para assegurar que a modelação está correta, realizamos os seguintes testes:

- **Verificação das restrições:** Todas as restrições impostas no modelo foram analisadas e comparadas com as definições matemáticas apresentadas no material de referência;
- **Análise da coerência dos resultados:** O resultado obtido foi comparado com soluções intuitivas para confirmar que a distribuição dos itens nos contentores minimizava o custo total.

2.6.2. Validação Computacional

Utilizámos o *LPSolve* para a resolução do modelo, analisando o *output* gerado e verificando que:

- Existe convergência para **soluções viáveis e ótimas** dentro do tempo esperado;
- São respeitadas as **restrições de capacidade** dos contentores;
- O plano de empacotamento respeita **a demanda dos itens** solicitados para empacotar;
- A soma dos comprimentos dos contentores (99) é **suficiente para empacotar todos os itens** (soma total do comprimento dos itens: 99);
- A solução ótima está **o mais minimizada possível**, pois é o valor mínimo que pode empacotar toda a demanda de itens. Que neste caso até corresponde ao próprio valor da soma total do comprimento dos itens.

A validação demonstrou que o modelo de fluxos em arcos gerou soluções consistentes com a formulação proposta.

3. Conclusão

Este trabalho permitiu explorar o problema de empacotamento proposto, aplicando técnicas de **programação linear e modelação por fluxos em arcos** para **otimizar** a disposição de itens em contentores com diferentes capacidades.

Durante o desenvolvimento do projeto, foi possível compreender a importância de uma **boa definição do modelo matemático e das suas restrições para a obtenção de soluções eficientes**. A implementação no ***LPSolve*** permitiu **validar o modelo** e garantir que os resultados obtidos **minimizavam a soma dos comprimentos dos contentores utilizados**, respeitando todas as condições do problema.

Trabalhos futuros poderão explorar melhorias na **eficiência computacional e na adaptação do modelo para problemas similares em aplicações reais**, como gestão de armazéns e logística.

Em suma, o projeto proporcionou uma **experiência valiosa na aplicação de técnicas de otimização linear**, fortalecendo o entendimento sobre a utilização de modelos matemáticos para resolver problemas práticos de engenharia.

4. Bibliografia

[1] J. M. Valério de Carvalho, «LP models for bin packing and cutting stock problems», *European Journal of Operational Research*, vol. 141, pp. 253–273, set. 2002.

[2] J. M. Valério de Carvalho, «Exact solution of bin-packing problems using column generation and branch-and-bound», *Annals of Operations Research*, vol. 86, pp. 629–659, jan. 1999.

[3] J. M. Valério de Carvalho, «Arc flow formulations based on dynamic programming: Theoretical foundations and applications», *European Journal of Operational Research*, vol. 296, pp. 3–21, jan. 2022.