

UNIVERSIDADE DO MINHO
LICENCIATURA EM ENGENHARIA INFORMÁTICA

INVESTIGAÇÃO OPERACIONAL

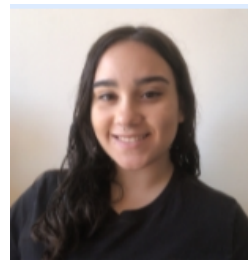
Trabalho Prático 1 *Bin Packing Problem*



Henrique Pereira
a100831



Luís Caetano
a100893



Mariana
Gonçalves
a100662



Maya Gomes
a100822

21 de maio de 2023

Conteúdos

1	Introdução	3
2	Questão 0: Indique o valor de xABCDE, e apresente duas tabelas como as do enunciado com os valores resultantes da aplicação da regra. Calcule também a soma dos comprimentos dos itens a empacotar.	3
3	Questão 1: Apresente a formulação deste problema	4
3.1	Formulação do problema	4
3.1.1	Descrição do Problema	4
3.1.2	Objetivo principal	4
3.1.3	Dados	4
3.2	Critérios de redução	5
3.2.1	Critério um:	6
3.2.2	Critério dois:	6
3.2.3	Critério três:	7
4	Questão 2: Apresente o modelo de programação linear	7
4.1	Variáveis de Decisão	7
4.2	Restrições	8
4.2.1	Restrições - Grupo I: Conservação de fluxo	10
4.2.2	Restrições - Grupo II: Itens	10
4.2.3	Restrições - Grupo III: Contentores	10
4.2.4	Declaração de Variáveis Inteiras	10
4.3	Função objetivo	11
5	Questão 3: Apresente o ficheiro de input	12
6	Questão 4: Apresente o ficheiro de output produzido pelo programa	13
7	Questão 5: Interprete a solução óptima, apresente o plano de empacotamento e indique a soma dos comprimentos dos contentores usados.	14
7.1	Interpretação da solução óptima	14
7.2	Apresentação do plano de empacotamento	14
7.3	Soma dos comprimentos dos contentores usados	15
8	Questão 6: Descreva os procedimentos usados para validar o modelo.	15
8.1	Validação intuitiva	15
8.2	Validação formal	15
8.2.1	grupo I de restrições	16
8.2.2	grupo II de restrições	16
8.2.3	grupo III de restrições	16
8.3	Aplicação na realidade	17

1 Introdução

Este relatório, no âmbito da unidade curricular de Investigação Operacional, contém os elementos fundamentais do primeiro trabalho prático proposto. O enunciado do trabalho propõe a criação de um modelo de programação linear para resolver um problema que envolve a colocação de um conjunto de itens em contentores.

Assim sendo, tem-se como objetivo principal fazer com que os contentores consigam empacotar os itens todos, reduzindo ao máximo a soma dos comprimentos dos contentores e não o número de contentores.

Os contentores (*bins*) têm uma capacidade W e uma lista de n itens de comprimento w_i , $0 \leq w_i \leq W$, com $i = 1, \dots, n$.

Para a resolução deste problema foi utilizado o *software* de programação linear *LPSolve*, muito conhecido neste âmbito de programação.

2 Questão 0: Indique o valor de xABCDE, e apresente duas tabelas como as do enunciado com os valores resultantes da aplicação da regra. Calcule também a soma dos comprimentos dos itens a empacotar.

Nesta questão, iremos proceder a elaboração do grafo de acordo com as indicações que se apresentam de seguida:

Considere-se xABCDE o número de inscrição do estudante do grupo com maior número de inscrição.

Tendo em conta a observação inicial e que os números mecanográficos dos estudantes têm a seguinte relação: $100662 < 100822 < 100831 < 100893$, iremos utilizar:

- xABCDE = 100893
- considerando que $k=bi$, temos que:
 - Como B é par, temos que $k1=0$;
 - Como C é par, temos que $k2= C+2=8+2=10$;
 - Como D é ímpar, temos que $k3=10$;
 - Como E é ímpar, temos que $k4=E+8=3+8=11$;
 - Quantidade disponível do contentor 2 = $B+1=0+1=1$;
 - Quantidade disponível do contentor 3 = $D+1=9+1=10$.

itens	Comprimento (w_i)	Quantidade (b_i)
1	1	0
2	2	10
3	3	10
4	4	11
5	5	5

Tabela 1: Aplicação do exemplo prático.

contentores	Comprimento (W_k)	Quantidade (B_k)
1	11	inf
2	10	1
3	7	10

Tabela 2: Aplicação do exemplo prático.

As fórmulas necessárias são dadas por:

- Capacidade Disponível: $\sum_k W_k B_k$
- Soma dos comprimentos dos itens: $\sum_i w_i b_i$

pelo que :

- número de *bins*: 3
- número de itens: 5
- $w_{\max} = 11$ (comprimento do maior)
- Capacidade Disponível = $11 \cdot \inf + 10 \cdot 1 + 7 \cdot 10 = 80 + 11 \cdot \inf$
- Soma dos comprimentos dos itens: $2 \cdot 10 + 3 \cdot 10 + 4 \cdot 11 + 5 \cdot 5 = 119$

3 Questão 1: Apresente a formulação deste problema

3.1 Formulação do problema

A formulação do problema é uma etapa que nos permite perceber o problema de um modo mais profundo. Desta forma, reuniram-se os dados, definiram-se as variáveis de decisão relevantes e o conjunto das restrições que se observam adiante. [3]

3.1.1 Descrição do Problema

O problema de empacotamento de itens em contentores representa uma classe de problemas designadas por *bin packing problems*.

Como referido no enunciado do trabalho, aborda-se um problema que engloba contentores de diferentes capacidades. Desta forma, é medida a eficiência com base na soma dos comprimentos dos itens dos contentores usados. As características dos contentores e itens dependem do valor de xABCDE (maior número de aluno do grupo) como anteriormente definido.

Este problema pode ser formulado como um problema de fluxo, pelo que, socorre-se de um grafo para a sua resolução. Deste modo, cada um dos vértices representa uma medida do comprimento possível e cada aresta representa uma alocação possível de um item de tamanho específico. [1] [2] [3]

3.1.2 Objetivo principal

O principal objetivo para a resolução deste problema é encontrar a sua solução ótima, ou seja, encontrar um conjunto contentores, de modo a otimizar a medida de eficiência. Desta forma, pretende-se determinar uma configuração ótima de alocação, utilizando o grafo em questão. A configuração encontrada deve respeitar o requisito de minimização da soma dos comprimentos dos itens alocados. Note-se que não se pretende minimizar o número de contentores.

3.1.3 Dados

Tendo em conta a formulação do problema sob a forma de um grafo considere-se os seguintes elementos do grafo (G):

- $G=(V,A)$, sendo que V representa os vértices e A as arestas;
- tem-se d vértices (com d entre 0 e W_{\max});
- cada aresta (d,e) vai corresponder a alocar um item, com um certo comprimento de e-d ($0 \leq d \leq e \leq W_{\max}$);
- cada aresta pode também significar porções vazias do BIN (perdas);

Para construir o grafo vamos seguir as seguintes regras:

- $A = \{(d, e) : 0 \leq d \leq e \leq W_{max} \wedge e - d = w_i, \text{ for } 1 \leq i \leq m\}$, o que significa que vamos ter arestas de um vértice menor para um vértice maior, em que a diferença dos vértices corresponde ao tamanho de um item. Assim sendo, teremos aqui tantas arestas quanto o número de itens.
- Arcos adicionais de perda: entre vértices adjacentes; $(d, d + 1), d \leq W_{max} - 1$.
- Finalmente, consideramos também arcos diretos entre o vértice que define a capacidade máxima de cada *bin* e o vértice inicial (0).

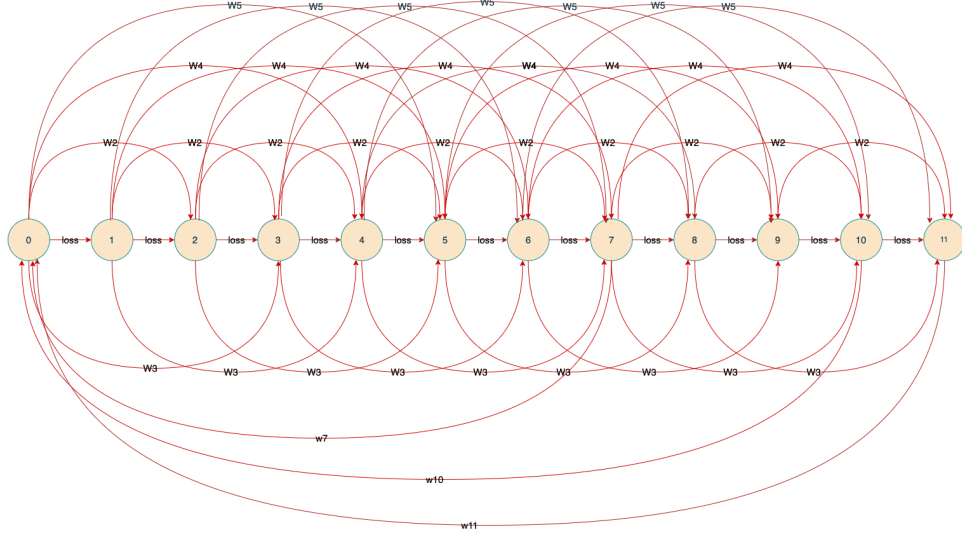


Figura 1: Grafo inicial.

3.2 Critérios de redução

O grafo resultante tem imensas arestas, portanto, é boa ideia aplicar alguns critérios de redução de arestas. A ideia destes critérios de redução é reduzir a simetria inerente ao grafo inicial, contudo, após a aplicação, ainda podem existir algumas soluções simétricas. De forma a tornar a aplicação dos critérios mais prática, realizou-se uma adaptação dos critérios, utilizando uma linguagem mais natural.

Considere-se o conjunto dos critérios seguintes[1]:

Criterion 1. *An arc $(k, k + w_e)$ of size w_e can only leave a node $k > 0$ if there is another arc $(k - w_d, k)$ of size $w_d \geq w_e$ entering k ; any node can leave node $k = 0$.*

Criterion 2. *All the loss arcs $(k, k + 1)$ can be removed for $k < w_m$ (recall that w_m is the smallest item).*

Criterion 3. *Given any node k that is the head of an arc of size w_d or $k = 0$, the only valid arcs for size w_e ($w_e < w_d$) are those that start at nodes $k + sw_e$, for $s = 0, 1, 2, \dots, b_e - 1$, with $k + (s + 1)w_e \leq W$, where b_e is the demand of items of size w_e .*

3.2.1 Critério um:

Neste critério, é descrito que para uma determinada aresta $(d, d + W_e)$ de tamanho W_e , só é possível sair do nó $d > 0$ se houver outra aresta $(d - W_d, d)$ de tamanho $W_d \geq W_e$ que entra no nó d e para além disso qualquer aresta pode sair do nó $d = 0$, independentemente de existirem outras arestas que entram ou saem desse nó.

Assim foram removidas as seguintes arestas:

- para $d=1$: $(1,3)$; $(1,4)$; $(1,5)$; $(1,6)$
- para $d=2$: $(2,5)$; $(2,6)$; $(2,7)$
- para $d=3$: $(3,7)$; $(3,8)$
- para $d=4$: $(4,9)$
- para $d=5$: nenhum
- para $d=6$: $(6,10)$; $(6,11)$
- para $d=7$: $(7,11)$
- para $d=8$: nenhum

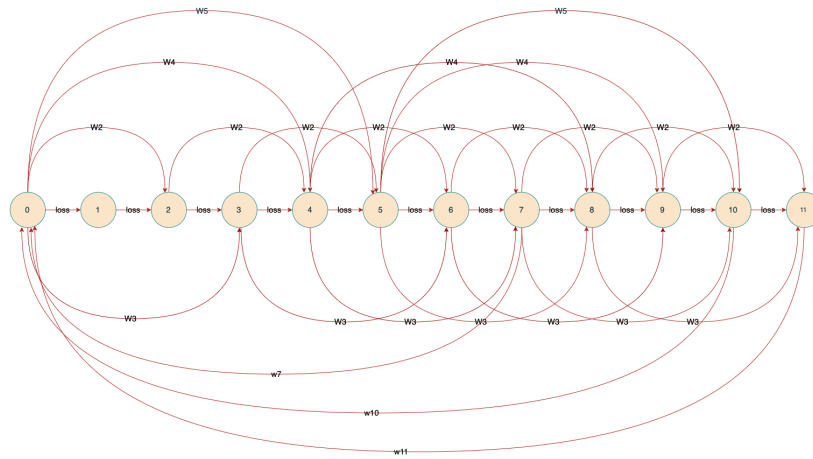


Figura 2: Critérios de remoção 1.

3.2.2 Critério dois:

Com base neste critério, temos que o tamanho do menor item é 2, isto é, $W_e=2$. Assim, todas as variáveis de perda, ou seja, os arcos da forma $(d, d+1)$ podem ser removidos se $d < W_e$. Desta forma, removemos os arcos $(0,1)$ e $(1,2)$.

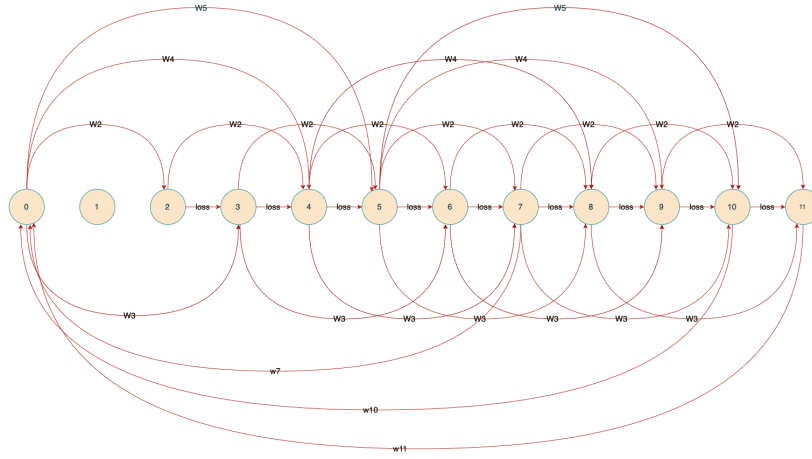


Figura 3: Critérios de remoção 2.

3.2.3 Critério três:

Este critério indica que, para um determinado nó d que é a cabeça de um arco de tamanho W_e ($W_e < W_d$), são aqueles que começam nos nós $d + S \cdot W_e$, para $S = 0, 1, 2, \dots, B_e - 1$, em que $d + (S + 1) \cdot W_e \leq W$, sendo que B_e é a quantidade de itens de tamanho W_e a serem entregues. Por outras palavras, este critério pretende reduzir o número consecutivo de arcos do mesmo tipo, ou seja, para cada item só pode existir um número consecutivo de arcos inferior a sua quantidade a alocar. Neste caso, não foi removido nenhum arco, por isso o grafo permanece inalterado.

4 Questão 2: Apresente o modelo de programação linear

4.1 Variáveis de Decisão

Variáveis de decisão são o conjunto de valores que se traduzem em decisões a implementar no sistema real. De acordo com a descrição do problema, sabemos de algumas variáveis descritas no enunciado tal como: o número de *bins*, o número de itens, a capacidade de cada *bins*, os comprimentos dos itens e as disponibilidades.

Como se pretende empacotar um conjunto de itens de comprimentos e disponibilidade variáveis, pressupõe-se que serão necessárias variáveis que representem esses fatores. Tendo em conta a abordagem, o comprimento de cada item seria representado por cada aresta com comprimento igual ao tamanho do item, o espaço alocado por um certo item seria representado entre dois vértices, cada um deles com um valor inteiro que significa o início da alocação e o fim da alocação, isto é, o espaço ocupado. Tendo em conta essa necessidade de empacotamento, foram necessárias variáveis:

- para representar cada aresta, utiliza-se x_{d_e} . Com d e e pertencentes a um domínio inteiro não negativo e finito (1).
- para representar o número de *bins* de capacidade W_d utiliza-se z_d (2);
- considera-se $d \neq e$;

Assim sendo, seguem-se as declarações do tipo variável:

x_{de} : d representa o vértice de origem e representa o vértice de destino. Permite representar uma aresta de (variável inteira).

$$\begin{aligned} d &\in \{0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\} \\ e &\in \{0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\} \\ x_{d_e} &\geq 0 \wedge \text{inteiro} \forall (d, e) \in A \end{aligned} \tag{1}$$

z_k : k representa a capacidade do *bin* k .

$$\begin{aligned} k &\in \{1, 2, 3\} \\ z_k &\geq 0 \wedge \text{inteiro} \end{aligned} \tag{2}$$

4.2 Restrições

Para obter o conjunto de restrições socorreu-se das seguintes considerações:

- existem três restrições :
 - restrições de conservação de fluxo (3);
 - restrições nos comprimentos de cada *bin* (4);
 - restrições relativas à quantidade disponível de cada *bin* (5).

Considere-se A' como o subconjunto de arcos resultantes após a aplicação dos critérios. A matriz anterior resume as restrições cuja a formulação matemática se segue:

$$-\sum_{(d,e) \in A'} x_{de} + \sum_{(e,f) \in A'} x_{ef} = \begin{cases} \sum_{k=1}^K z_k & \text{if } e = 0, \\ -z_k & \text{for } e = W_k, \quad k = 1, \dots, K, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

$$\sum_{(d,d+w_i) \in A'} x_{d,d+w_i} \geq b_i, \quad i = 1, \dots, m, \tag{4}$$

$$z_k \leq B_k, \quad k = 1, \dots, K, \tag{5}$$

De forma a definirmos as diversas restrições foi criada uma matriz com base no nosso grafo final colocando 1 nos arcos que entram e -1 nos arcos que saem.

Tabela 3: Arc flow model.

As próximas secções destinam-se a elucidar os vários grupos de restrições.

4.2.1 Restrições - Grupo I: Conservação de fluxo

Estas restrições garantem que é respeitado o fluxo representado no grafo apresentado, ou seja, que o fluxo de entrada é igual ao fluxo de saída (3).

```
//grupo I : conservação de fluxo
NODE0: z1 + z2 + z3 - x0_5 - x0_4 - x0_3 - x0_2 = 0;
//NODE1: não existe
NODE2: x0_2 - x2_4 - x2_3 = 0;
NODE3: x0_3 - x3_6 - x3_5 + x2_3 - x3_4 = 0;
NODE4: x0_4 - x4_8 - x4_7 + x2_4 - x4_6 + x3_4 - x4_5 = 0;
NODE5: x0_5 - x5_10 - x5_9 + x3_5 - x5_7 + x4_5 - x5_6 = 0;
NODE6: x3_6 - x6_9 + x4_6 - x6_8 + x5_6 - x6_7 = 0;
NPDE7: -z3 + x4_7 - x7_10 + x5_7 - x7_9 + x6_7 - x7_8 = 0;
NODE8: x4_8 - x8_11 + x6_8 - x8_10 + x7_8 - x8_9 = 0;
NODE9: x5_9 + x6_9 + x7_9 - x9_11 + x8_9 - x9_10 = 0;
NODE10: -z2 + x5_10 + x7_10 + x8_10 + x9_10 - x10_11 = 0;
NODE11: -z1 + x8_11 + x9_11 + x10_11 = 0;
```

Figura 4: Restrição Grupo I.

4.2.2 Restrições - Grupo II: Itens

Esta restrição serve para garantir que a quantidade procurada dos itens é satisfeita (4).

```
// grupo II : Itens
//ITEM1: x2_3 + x3_4 + x4_5 + x5_6 + x6_7 + x7_8 + x8_9 + x9_10 + x10_11 >= 0; (não existe)
ITEM2: x0_2 + x2_4 + x3_5 + x4_6 + x5_7 + x6_8 + x7_9 + x8_10 + x9_11 >=10;
ITEM3: x0_3 + x3_6 + x4_7 + x6_9 + x7_10 + x8_11 >=10;
ITEM4: x0_4 + x4_8 + x5_9 >=11;
ITEM5: x0_5 + x5_10 >=5;
```

Figura 5: Restrição Grupo II.

4.2.3 Restrições - Grupo III: Contentores

Esta restrição garante que o fluxo no arco de retorno de um dado contentor está limitado pelo número de *bins* disponíveis para essa capacidade (5).

```
// grupo II : Contentores
BIN1: z1 <= 30000;
BIN2: z2 <=1;
BIN3: z3 <=10;
```

Figura 6: Restrição Grupo III.

4.2.4 Declaração de Variáveis Inteiras

Finalmente, temos ainda de declarar as variáveis como inteiras.

```
// variáveis inteiras
int
z1, z2, z3
x0_5, x5_10,
x0_4, x4_8, x5_9,
x0_3, x3_6, x4_7, x6_9, x7_10, x8_11,
x0_2, x2_4, x3_5, x4_6, x5_7, x6_8, x7_9, x8_10, x9_11,
x2_3, x3_4, x4_5, x5_6, x6_7, x7_8, x8_9, x9_10, x10_11;
```

Figura 7: Variáveis Inteiras.

4.3 Função objetivo

Não esquecendo nunca o objetivo do problema - determinar as arestas a adicionar para calcular a soma dos comprimentos dos contentores mais adequada, criamos a nossa função objetivo.

A função objetivo serve para minimizar as restrições em relação a alocação que anteriormente definimos (6).

$$\min \sum_{k=1}^K W_k z_k \quad (6)$$

```
// Função objetivo
min: 11 z1 + 10 z2 + 7 z3;
```

Figura 8: Função objetivo.

O modelo pode então resumir se a:

$$\begin{aligned} \min \quad & \sum_{k=1}^K W_k z_k \\ \text{s.t.} \quad & - \sum_{(d,e) \in A'} x_{de} + \sum_{(e,f) \in A'} x_{ef} = \begin{cases} \sum_{k=1}^K z_k & \text{if } e = 0, \\ -z_k & \text{for } e = W_k, \quad k = 1, \dots, K, \\ 0 & \text{otherwise,} \end{cases} \\ & \sum_{(d,d+w_i) \in A'} x_{d,d+w_i} \geq b_i, \quad i = 1, \dots, m, \\ & z_k \leq B_k, \quad k = 1, \dots, K, \\ & x_{de} \geq 0 \text{ and integer } \quad \forall (d,e) \in A', \\ & z_k \geq 0 \text{ and integer, } \quad k = 1, \dots, K. \end{aligned}$$

5 Questão 3: Apresente o ficheiro de input

Depois de percebermos o problema e de o formularmos corretamente através da programação linear, procedemos ao software LPSolve, para encontrar a respetiva solução ótima no contexto do problema.

```
1 // ficheiro LPSOLVE
2 // Função objetivo
3 min: 11 z1 + 10 z2 + 7 z3;
4
5 //restrições
6 //grupo I : conservação de fluxo
7 NODE0: z1 + z2 + z3 - x0_5 - x0_4 - x0_3 - x0_2 = 0;
8 //NODE1: não existe
9 NODE2: x0_2 - x2_4 - x2_3 = 0;
10 NODE3: x0_3 - x3_6 - x3_5 + x2_3 - x3_4 = 0;
11 NODE4: x0_4 - x4_8 - x4_7 + x2_4 - x4_6 + x3_4 - x4_5 = 0;
12 NODE5: x0_5 - x5_10 - x5_9 + x3_5 - x5_7 + x4_5 - x5_6 = 0;
13 NODE6: x3_6 - x6_9 + x4_6 - x6_8 + x5_6 - x6_7 = 0;
14 NODE7: - z3 + x4_7 - x7_10 + x5_7 - x7_9 + x6_7 - x7_8 = 0;
15 NODE8: x4_8 - x8_11 + x6_8 - x8_10 + x7_8 - x8_9 = 0;
16 NODE9: x5_9 + x6_9 + x7_9 - x9_11 + x8_9 - x9_10 = 0;
17 NODE10: -z2 + x5_10 + x7_10 + x8_10 + x9_10 - x10_11 = 0;
18 NODE11: -z1 + x8_11 + x9_11 + x10_11 = 0;
19
20
21 // grupo II : Itens
22
23
24 //ITEM1: x2_3 + x3_4 + x4_5 + x5_6 + x6_7 + x7_8 + x8_9 + x9_10 + x10_11 >= 0; (não existe)
25 ITEM2: x0_2 + x2_4 + x3_5 + x4_6 + x5_7 + x6_8 + x7_9 + x8_10 + x9_11 >=10;
26 ITEM3: x0_3 + x3_6 + x4_7 + x6_9 + x7_10 + x8_11 >=10;
27 ITEM4: x0_4 + x4_8 + x5_9 >=11;
28 ITEM5: x0_5 + x5_10 >=5;
29
30 // grupo III : Contentores
31
32 BIN1: z1 <= 30000;
33 BIN2: z2 <=1;
34 BIN3: z3 <=10;
35
36
37 // variáveis inteiras
38 int
39 z1, z2, z3
40 x0_5, x5_10,
41 x0_4, x4_8, x5_9,
42 x0_3, x3_6, x4_7, x6_9, x7_10, x8_11,
43 x0_2, x2_4, x3_5, x4_6, x5_7, x6_8, x7_9, x8_10, x9_11,
44 x2_3, x3_4, x4_5, x5_6, x6_7, x7_8, x8_9, x9_10, x10_11;
45
```

Figura 9: Ficheiro de input.

6 Questão 4: Apresente o ficheiro de output produzido pelo programa

```
Value of objective function: 119.00000000

Actual values of the variables:
z1      8
z2      1
z3      3
x0_5    5
x0_4    6
x0_3    1
x0_2    0
x2_4    0
x2_3    0
x3_6    1
x3_5    0
x3_4    0
x4_8    0
x4_7    6
x4_6    0
x4_5    0
x5_10   0
x5_9    5
x5_7    0
x5_6    0
x6_9    1
x6_8    0
x6_7    0
x7_10   1
x7_9    2
x7_8    0
x8_11   0
x8_10   0
x8_9    0
x9_11   8
x9_10   0
x10_11  0
```

Figura 10: Ficheiro de Output no Terminal.

7 Questão 5: Interprete a solução ótima, apresente o plano de empacotamento e indique a soma dos comprimentos dos contentores usados.

7.1 Interpretação da solução ótima

Através da matriz estabelecida, conseguimos criar o ficheiro LPSolve com o qual se obteve a solução ótima. A solução ótima que corresponde a forma mais reduzida da soma dos comprimentos dos contentores.

Logo, a soma dos comprimentos dos contentores que devemos alocar é igual a 119 unidades. Com base nisso, obteve-se o gráfico com os diversos percursos, isto é, as diferentes alocações da solução ótima.

7.2 Apresentação do plano de empacotamento

Com base no ficheiro output de configuração ótima obtido através do LPSolve, fomos desenhando o conjunto de caminhos da solução ótima, sendo que cada caminho representa uma alocação.

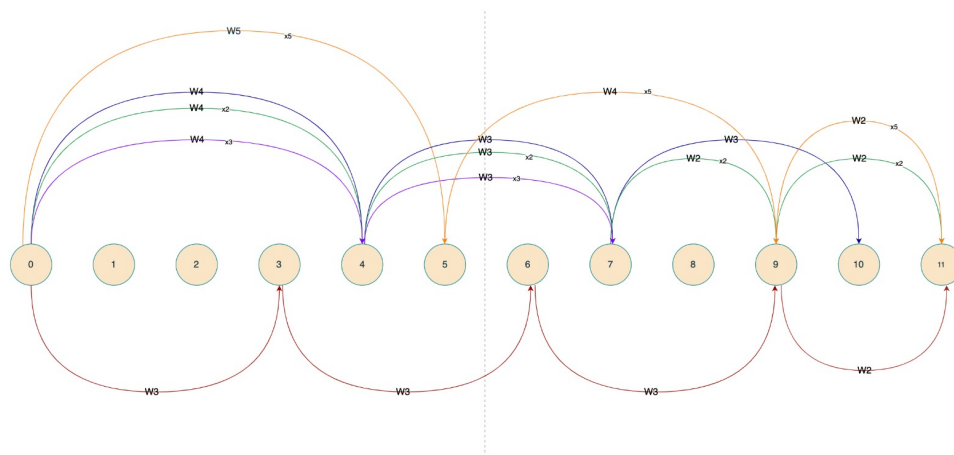


Figura 11: Solução ótima.

Os diversos caminhos obtidos são :

- (0,3) -> (3,6) -> (6,9) -> (9,11) -> (11,0);
 - capacidade = 11; $z1 - 1 = 8 - 1 = 7$;
 - representado a vermelho na figura 12;
- (0,4) -> (4,7) -> (7,10) -> (10,0);
 - capacidade = 10; $z2 - 1 = 1 - 1 = 0$;
 - representado a azul na figura 12;
- (0,4) -> (4,7) -> (7,9) -> (9,11) -> (11,0);
 - temos dois caminhos desse;
 - capacidade = 11; $z1 - 2 = 7 - 2 = 5$;
 - representado a verde na figura 12;
- (0,4) -> (4,7) -> (7,0);
 - temos três caminhos desse;
 - capacidade = 7; $z3 - 3 = 3 - 3 = 0$;
 - representado a roxo na figura 12;
- (0,5) -> (5,9) -> (9,11) -> (11,0);

- temos cinco caminhos desse;
- capacidade = 11; $z1 - 4 = 5 - 5 = 0$;
- representado a laranja na figura 12;

7.3 Soma dos comprimentos dos contentores usados

No início do processo, foi calculada a soma dos comprimentos dos itens (ponto 2). Desta forma, concluímos desde o início que a soma dos comprimentos dos contentores que devemos alocar é igual a 119.

Notamos que o ficheiro output apresenta nos um parâmetro "Value of objective function: 119.0". Sendo que o valor é igual, podemos concluir a veracidade do nosso ficheiro output.

8 Questão 6: Descreva os procedimentos usados para validar o modelo.

8.1 Validação intuitiva

Para realizarmos uma validação intuitiva do nosso modelo escolhemos responder a algumas perguntas:

- **O output alocou todos os itens?**

Com base nas quantidades de cada itens calculadas no início do trabalho, temos, no total, $10+10+11+5 = 36$ itens. Se somarmos as quantidades que nos aparecem no ficheiro output temos $5+6+1+1+6+5+1+1+2+8=36$ itens. Desta forma, todos os itens foram alocados.

- **O output usou corretamente a capacidade de cada Bin?**

Com base nos dados iniciais sabemos que o Bin nº1 tem capacidade infinita, o Bin nº2 tem capacidade 1 e o Bin nº3 tem capacidade 10. O ficheiro output afirma que temos 8 $z1$, 1 $z2$ e 3 $z3$, sendo as capacidades respeitadas.

- **O output alocou corretamente os itens de acordo com o comprimento?**

Dados os resultados obtidos no LpSolve a função objetivo tem o valor de 119.0 algo que se comprova através do cálculo da soma dos comprimentos dos itens anteriormente calculado.

- **O output respeitou os critérios de redução?**

O output respeitou os critérios de redução.

Relativamente ao critério 1 sabemos que de onde sai um arco de um determinado valor, neste só podem entrar arcos maiores ou iguais, o que se verifica no grafo da solução ótima.

O critério 2 afirma que arcos da forma $(d,d+1)$ podem ser removidos caso $d < w_n$, o que verificamos na solução ótima.

O critério 3 relaciona-se com a quantidade de cada item, sendo também respeitado na solução ótima, pois temos:

- nºde arcos $w2 = 10 =$ quantidade de itens 2 = 10
- nºde arcos $w3 = 10 =$ quantidade de itens 3 = 10
- nºde arcos $w4 = 11 =$ quantidade de itens 4 = 11
- nºde arcos $w5 = 1 <$ quantidade de itens 5 = 5

8.2 Validação formal

Para a validação formal iremos recorrer a substituição dos valores do ficheiro LPSolve com os valores obtidos no output e verificar se as equações mantêm se verdadeiras.

8.2.1 grupo I de restrições

- NODE0: $z1 + z2 + z3 - x0_5 - x0_4 - x0_3 - x0_2 = 8 + 1 + 3 - 5 - 6 - 1 - 0 = 0 \equiv Verdadeira;$
- NODE1: não existe;
- NODE2: $x0_2 - x2_4 - x2_3 = 0 - 0 - 0 = 0 \equiv Verdadeira;$
- NODE3: $x0_3 - x3_6 - x3_5 + x2_3 - x3_4 = 1 - 1 - 0 - 0 - 0 = 0 \equiv Verdadeira;$
- NODE4: $x0_4 - x4_8 - x4_7 + x2_4 - x4_6 + x3_4 - x4_5 = 6 - 0 - 6 + 0 - 0 + 0 - 0 = 0 \equiv Verdadeira;$
- NODE5: $x0_5 - x5_10 - x5_9 + x3_5 - x5_7 + x4_5 - x5_6 = 5 - 0 - 5 + 0 - 0 + 0 - 0 = 0 \equiv Verdadeira;$
- NODE6: $x3_6 - x6_9 + x4_6 - x6_8 + x5_6 - x6_7 = 1 - 1 - 0 - 0 + 0 - 0 = 0 \equiv Verdadeira;$
- NODE7: $-z3 + x4_7 - x7_10 + x5_7 - x7_9 + x6_7 - x7_8 = -3 + 6 - 1 + 0 - 2 + 0 - 0 = 0 \equiv Verdadeira;$
- NODE8: $x4_8 - x8_11 + x6_8 - x8_10 + x7_8 - x8_9 = 0 - 0 + 0 - 0 + 0 - 0 = 0 \equiv Verdadeira;$
- NODE9: $x5_9 + x6_9 + x7_9 - x9_11 + x8_9 - x9_10 = 5 + 1 + 2 - 8 + 0 - 0 = 0 \equiv Verdadeira;$
- NODE10: $-z2 + x5_10 + x7_10 + x8_10x9_10 - x10_11 = -1 + 0 + 1 + 0 + 0 - 0 = 0 \equiv Verdadeira;$
- NODE11: $-z1 + x8_11 + x9_11 + x10_11 = -8 + 0 + 8 + 0 = 0 \equiv Verdadeira;$

8.2.2 grupo II de restrições

- ITEM1: não existe;
- ITEM2: $x0_2 + x2_4 + x3_5 + x4_6 + x5_7 + x6_8 + x7_9 + x8_10 + x9_11 = 0 + 0 + 0 + 0 + 0 + 2 + 0 + 8 = 10 > = 10 \equiv Verdadeira;$
- ITEM3: $x0_3 + x3_6 + x4_7 + x6_9 + x7_10 + x8_11 = 1 + 1 + 6 + 1 + 1 + 0 = 10 > = 10 \equiv Verdadeira;$
- ITEM4: $x0_4 + x4_8 + x5_9 = 6 + 0 + 5 = 11 > = 11 \equiv Verdadeira;$
- ITEM5: $x0_5 + x5_10 = 5 + 0 = 5 > = 5 \equiv Verdadeira;$

8.2.3 grupo III de restrições

- BIN1 : $z1 = 8 < = 30000 \equiv Verdadeira;$
- BIN2 : $z2 = 1 < = 1 \equiv Verdadeira;$
- BIN3 : $z3 = 3 < = 10 \equiv Verdadeira;$

8.3 Aplicação na realidade

Para além disso, decidimos representar o empacotamento da solução ótima num esquema tal como acaba por ser sugerido no enunciado. Assim, acabamos por esquematizar a solução utilizando uma representação mais representativa da aplicação do problema na vida real.

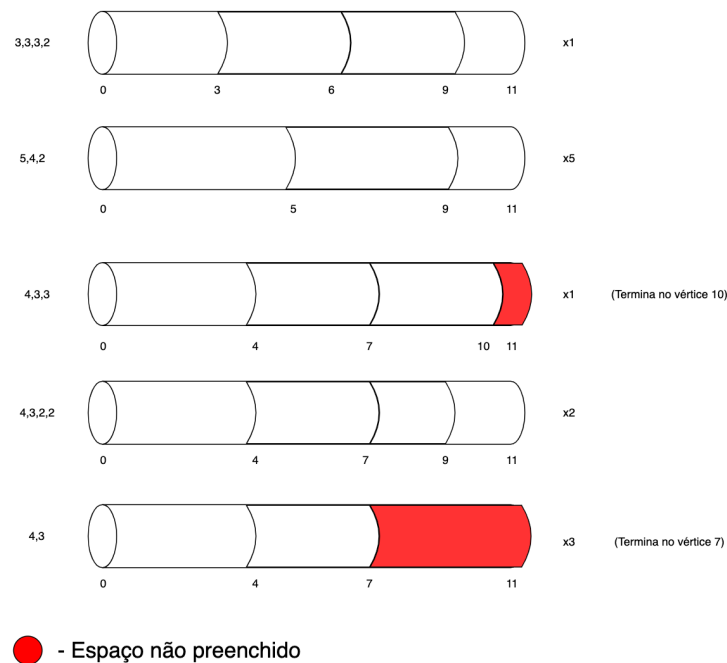


Figura 12: Esquema de empacotamento.

Referências

- [1] Brandão, F., Pedroso, J.P., Pedroso, P.: Solving bin packing related problems using an arc flow formulation, <http://www.dcc.fc.up.pt/Pubs/>
- [2] Gradišar, D., Glavan, M.: Material requirements planning using variable-sized bin-packing problem formulation with due date and grouping constraints. Processes **8**, 1–16 (10 2020). <https://doi.org/10.3390/pr8101246>
- [3] Val, J.M., Carvalho, E.D.: Lp models for bin packing and cutting stock problems, www.elsevier.com/locate/dsw