Universidade do Minho
Escola de Engenharia

Investigação Operacional

Parte 1
LEI - Universidade do Minho

Março de 2023

Flávia Araújo - A96587
Inês Marques - A100606
Margarida Pimenta - A100830
Margarida Lopes - A100834

0. Considerações Iniciais

Neste problema é disponibilizado um certo número de contentores de igual capacidade, e pretende-se empacotar um conjunto de itens nesses contentores de maneira eficiente. É possível ter contentores com algum espaço não ocupado, ou até ter contentores vazios, porém não é possível exceder a capacidade dos mesmos.

De acordo com o maior número de estudante do nosso grupo (100834), o número de contentores disponíveis de cada comprimento e o número de itens a empacotar de cada comprimento são dados pelas seguintes tabelas:

Contentores:

Comprimento	Quantidade disponível
11	ilimitada
10	1
7	4

Itens:

Comprimento	Quantidade
1	0
2	10
3	10
4	6
5	5

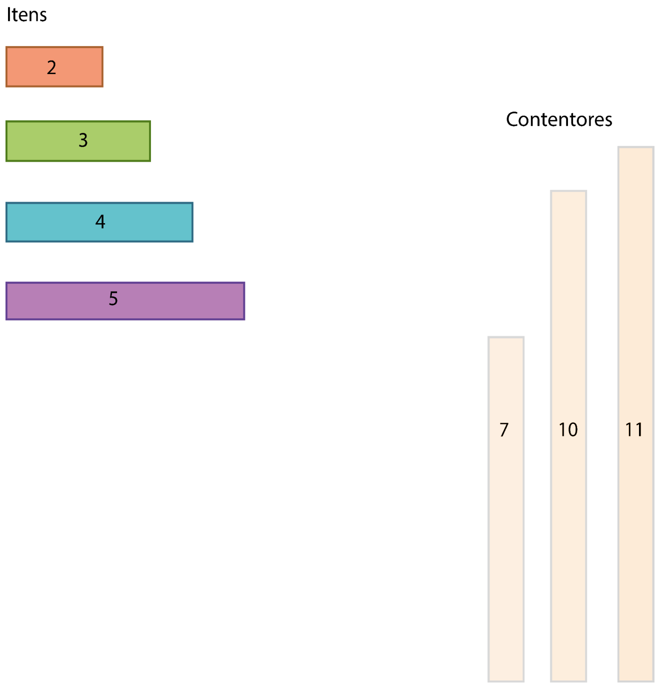

Fig. 1.1: Representação gráfica dos contentores, itens e os seus respectivos comprimentos.

1. Formulação do Problema

O *bin packing problem* consiste em acomodar um conjunto de itens num número limitado de contentores, minimizando o número de contentores utilizados, bem como tendo em conta que o espaço de sobra no contentor seja mínimo. Este tipo de problemas seguem um raciocínio similar aos *cutting stock problems,* com os quais tivemos contacto nas aulas práticas.

Neste caso, tal como foi explicado no ponto anterior, existem 3 tipos diferentes de contentores com diferentes comprimentos (a primeira restrição). Além disso, há uma quantidade de itens de comprimentos variados, e o seu empacotamento nos contentores tem que respeitar o comprimento dos mesmos (a segunda restrição).

Tendo em conta os dados do problema, delimita-se o objetivo como sendo o seguinte: *qual é o padrão mais adequado a cada contentor para garantir que todos os itens são empacotados utilizando um comprimento mínimo e sem ultrapassar o número de contentores?*

Para calcular os padrões, soma-se os comprimentos dos itens que irão ser empacotados num certo contentor, e ter em atenção para não só este valor não superar o comprimento do contentor, como também ter em conta para o espaço de sobra ser mínimo (assim teremos a certeza que é um modelo ótimo).

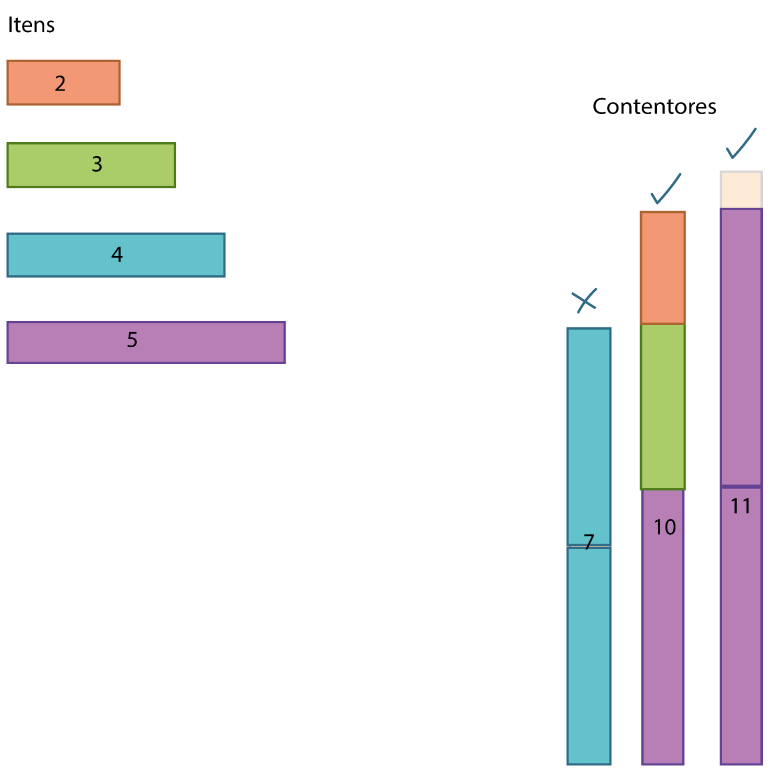

Fig.1.2: *Bin packing problem* (exemplo).

Utilizando a figura 1, observa-se que é possível haver contentores com algum espaço livre, como o contentor de comprimento 11 e alguns completamente cheios, como o contentor de comprimento 10. No caso do contentor de comprimento 7, são utilizados dois itens de comprimento 4 cada, o que excede o limite do contentor - o que está fora de questão.

Posto isto, a formulação do problema consiste em uma função objetivo *min* e duas restrições - sendo a primeira a soma de todos os padrões, para cada contentor, não pode ultrapassar o número de contentores, e a segunda que todos os itens têm que estar em contentores. Este problema é baseado na variável de decisão xij, em que i é o tipo de contentor e j é o número do padrão. A solução ótima vai corresponder ao menor comprimento usado.

2. Modelo de Programação Linear

2.1. Variáveis de decisão

xij: i é o tipo de contentor (de comprimento 7, 10 ou 11) e j é o número do padrão de empacotamento, com i $\in \{1, 2, 3\}$ e j $\in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$.

$x11, x12, x13, x14, x15, x16, x21, x22, x23, x24, x25, x26, x27, x28, x29, x210, x211, x212, x31, x32, x33, x34, x35, x36, x37, x38, x39, x310, x311, x312, x313, x314 \geq 0.$

2.2. Parâmetros

Existe 3 tipos de contentores disponíveis:

1. Contentores de comprimento 7 (4);
2. Contentores de comprimento 10 (1);
3. Contentores de comprimento 11 (ilimitado).

E 4 tipos de itens, de comprimentos 2, 3, 4 e 5, e o comprimento disponível no contentor i utilizando o padrão de empacotamento j. Os comprimentos utilizados em cada contentor segundo padrões de empacotamento diferentes podem ser calculados a partir das seguintes tabelas:

	x1	x2	x3	x4	x5	x6
2	1	0	1	2	3	0
3	0	1	0	1	0	2
4	0	1	1	0	0	0
5	1	0	0	0	0	0
Total:	7	7	6	7	6	6

Fig 2.1: Padrões de empacotamento e comprimento utilizado (contentores tipo 1).

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12
2	0	5	1	0	0	0	1	1	3	2	3	2
3	0	0	0	3	0	2	1	1	0	2	1	0
4	0	0	2	0	1	1	1	0	1	0	0	0
5	2	0	0	0	1	0	0	1	0	0	0	1
Total:	10	10	10	9	9	10	9	10	10	10	9	9

Fig 2.2: Padrões de empacotamento e comprimento utilizado (contentores tipo 2).

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14
2	0	5	1	1	1	0	2	1	3	2	4	3	0	0
3	0	0	0	3	0	2	1	1	0	2	1	0	2	1
4	0	0	2	0	1	1	1	0	1	0	0	0	0	2
5	2	0	0	0	1	0	0	1	0	0	0	1	1	0
Total:	10	10	10	11	11	10	11	10	10	10	11	11	11	11

Fig 2.3: Padrões de empacotamento e comprimento utilizado (contentores tipo 3).

2.3. Função Objetivo

Minimização do comprimento utilizado de contentores para empacotar os itens:

$$min: \sum_i (\sum_j di \times xij)$$

Em que *di* corresponde ao tipo de contentor utilizado (comprimento 7, 10 ou 11), e *xij* à quantidade de contentores do tipo i que foram colocados no padrão j.

2.4. Restrições

Tal como foi referido no ponto 1, a primeira restrição delimita a quantidade de contentores disponíveis de cada tamanho. Com isto, não se pode utilizar mais que 4 contentores de comprimento 7 e mais que 1 contentor de comprimento 10. À exceção dos contentores de comprimento 11, cujo a sua quantidade é ilimitada, não há restrição (fica apenas que tem que ser maior ou igual a 0).

A segunda restrição garante que todos os itens sejam colocados em contentores. Para isso, soma-se a quantidade de cada tipo em todos os padrões. No final, os somatórios devem ser pelo menos o número de unidades do respectivo tipo de item.

Nesta restrição, o sinal maior ou igual (">=") impede que um padrão não seja escolhido devido à sua composição exigir um item cujas unidades já foram empacotadas. Embora isso não seja uma solução ideal, deve ser contabilizado.

```
/* Objective function */
min: 7 x11+7 x12+ 7 x13+ 7 x14+ 7 x15+ 7 x16+
     10 x21+ 10 x22+ 10 x23+ 10 x24+ 10 x25+ 10 x26+ 10 x27+ 10 x28+ 10 x29+ 10 x210+ 10 x211 + 10 x212+
     11 x31+ 11 x32+ 11 x33+ 11 x34+ 11 x35+ 11 x36+ 11 x37+ 11 x38+ 11 x39+ 11 x310+ 11 x311+ 11 x312+ 11 x313+ 11 x314;

/* Variable bounds */

Contentor_7  : x11+x12+x13+x14+x15+x16<=4;
Contentor_10 : x21+x22+x23+x24+x25+x26+x27+x28+x29+x210+x211+x212<=1;
Contentor_11 : x31+x32+x33+x34+x35+x36+x37+x38+x39+x310+x311+x312+x313+x314 >=0;

I2: x11 + x13 + 2 x14 + 3 x15 + 5 x22 +   x23 + x27 + x28 + 3 x29  + 2 x210 + 3 x211+ 2 x212 + 5 x32 +   x33 + x34 + x35  + 2 x37  +   x38  + 3 x39  + 2 x310 + 4 x31 + 3 x312 >= 10;
I3: x12 + x14 + 2 x16          + 3 x24 + 2 x26 + x27 + x28 + 2 x210 +   x211                    + 3 x34 + 2 x36 + x37 + x38  + 2 x310 +   x311 + 2 x313 +   x314 >=10;
I4: x12 + x13                  + 2 x23 +   x25 + x26 + x27 +   x29                              + 2 x33 +   x35 + x36 + x37  +   x39  + 2 x314 >=6;
I5: x11                        + 2 x21 +   x25 + x28 + x212                                     + 2 x31 +   x35 + x38 + x312 +   x313 >=5;

int x11,x12,x13,x14,x15,x16;
int x21,x22,x23,x24,x25,x26,x27,x28,x29,x210,x211,x212;
int x31,x32,x33,x34,x35,x36,x37,x38,x39,x310,x311,x312,x313,x314;
```

Fig.3: Ficheiro de *input* no *LPSolve IDE*.

3. Solução Ótima

Após introduzir o modelo do problema no *LPSolve,* chega-se a uma aparente solução ótima.

Variables	MILP ...	MILP ...	MILP ...	MILP ...	result
	108	104	101	99.00...	99.00...
x11	3	3	1	0	0
x12	0	0	1	0	0
x13	0	0	0	0	0
x14	0	1	0	0	0
x15	0	0	0	0	0
x16	0	0	0	0	0
x21	1	1	1	0	0
x22	0	0	0	0	0
x23	0	0	0	0	0
x24	0	0	0	0	0
x25	0	0	0	0	0
x26	0	0	0	0	0
x27	0	0	0	0	0
x28	0	0	0	0	0
x29	0	0	0	0	0
x210	0	0	0	0	0
x211	0	0	0	0	0
x212	0	0	0	0	0
x31	0	0	0	0	0
x32	0	0	0	0	0
x33	3	3	1	0	0
x34	4	3	3	0	0
x35	0	0	2	0	0
x36	0	0	0	0	0
x37	0	0	0	0	0
x38	0	0	0	0	0
x39	0	0	1	0	0
x310	0	0	0	0	0
x311	0	0	0	1	1
x312	0	0	0	2	2
x313	0	0	0	3	3
x314	0	0	0	3	3

Constraints	MILP ...	MILP ...	MILP ...	MILP ...	result
	108	104	101	99.00...	99.00...
Contentor_7	3	4	2	0	0
Contentor_10	1	1	1	0	0
Contentor_11	7	6	7	9.000...	9.000...
I2	10	11	10	10	10
I3	12	10	10	10	10
I4	6	6	6	6	6
I5	5	5	5	5.000...	5.000...

Fig.4: *Output* produzido pelo LPSolve IDE.

Como observável no ficheiro de *output*, o comprimento total dos contentores usados é 99. Posto isto, são precisos 9 contentores de comprimento 11, utilizando os seguintes padrões de empacotamento para obter a solução ótima: 1 $x311$, 2 $x312$, 3 $x313$ e 3 $x314$.

Do mesmo modo, pela figura 4, observa-se que o número de itens empacotados foi 31 - de entre estes, 10 são de comprimento 2, 10 são de comprimento 3, 6 de comprimento 4 e 5 de comprimento 5 -, que corresponde ao número de itens para empacotar. De igual modo, o número de contentores utilizados também é respeitado. Sendo assim, conclui-se que todas as restrições são devidamente respeitadas.

4. Validação do Modelo

Nesta fase, para validar um modelo faz-se uma análise às restrições, se estão a ser cumpridas, e à solução ótima proposta, se aproxima-se do somatório dos itens.

Sendo assim, como verifica-se no ponto anterior, as restrições são respeitadas, tanto as dos itens, quanto as dos contentores. Além disso, na comparação da solução ótima com o somatório dos itens é verificável que são iguais, o que reforça a tese de que obteve-se a melhor solução.

Com base nestas observações, conclui-se então que este modelo é válido e a solução é ótima.

5. Considerações Finais

Graças à resolução e explicação do problema de corte abordado nas aulas práticas, tornou-se mais fácil e tangível a resolução deste problema para o nosso nível de conhecimento.

Contudo, surgiram algumas dificuldades e inseguranças. Por exemplo, a solução ótima que apenas usa contentores com comprimento 11 levantou inseguranças em relação à eficácia do modelo, além das dificuldades em expor e explicar as ideias do modelo.

Em síntese, o trabalho foi acessível e simples, embora com um certo grau de desafio.