23-03-2024

Trabalho 1

Investigação Operacional

Ana Sá Oliveira a104437
Inês Silva Marques a104263
José Rafael De Oliveira Vilas Boas a76350
Tomás Pinto Rodrigues a104448

Índice

Introdução

Neste trabalho iremos resolver um problema de empacotamento de itens, de diversos tamanhos, em contentores, também de diversos tamanhos. Para resolver este problema iremos utilizar o modelo de 'úm-corte' de Dyckhoff, designado na literatura anglo-saxónica por one-cut model. Primeiro, vamos obter os dados do problema e depois iremos formular o problema, construir um modelo de programação linear, colocar este modelo num solver, obter a solução ótima, interpretá-la e representá-la, e terminaremos por validar o modelo.

Determinação dos dados do problema

O número 104448 é o número de inscrição do estudante do grupo com maior número de inscrição, logo o valor de xABCDE é 104448.

xABCDE = 104448

x = 1

A = 0

B = 4

C = 4

D = 4

E = 8

Como B=4, B+1 = 4+1 = 5, logo temos disponíveis 5 contentores de comprimento 10. Como D=4, D+1 = 4+1 = 5, logo temos disponíveis 5 contentores de comprimento 7.

Tabela 1 – O número de contentores disponíveis de cada comprimento.

contentores	
comprimento	quantidade disponível
11	ilimitada
10	5
7	5

Como B=4, B é par, logo k1=0, ou seja, existem 0 itens de comprimento 1 para empacotar. Como C=4, C é par, logo k2 = C+2 = 4+2 = 6, ou seja, existem 6 itens de comprimento 2 para empacotar. Como D=4, D é par, logo k3=0, ou seja, existem 0 itens de comprimento 3 para empacotar. Como E=8, E é par, logo k4= E+2 = 8+2 = 10, ou seja, existem 10 itens de comprimento 4 para empacotar.

Tabela 2 – O número de itens a empacotar de cada comprimento.

Itens	
comprimento	quantidade
1	0
2	6
3	0
4	10
5	5

A soma dos comprimentos dos itens a empacotar é igual a 77.

i – Comprimento de um item.

q_i – Quantidade de itens a empacotar de comprimento i.

$$S = \sum_{i=1}^{5} i * q_i = 1 * 0 + 2 * 6 + 3 * 0 + 4 * 10 + 5 * 5 = 2 * 6 + 4 * 10 + 5 * 5 = 77$$

Formulação do problema

Descrição do problema

O problema que iremos abordar baseia-se nos dados calculados anteriormente. Trata-se de um problema de empacotamento com contentores de diferentes tamanhos. Os recursos disponíveis são 5 contentores de comprimento 7, 5 contentores de comprimento 10, e uma quantidade ilimitada de contentores de comprimento 11. Os itens a empacotar são 6 itens de comprimento 2, 10 itens de comprimento 4 e 5 itens de comprimento 5. Uma solução admissível do problema teria de empacotar todos os itens sem exceder a capacidade dos contentores nem a quantidade disponível de cada um.

Objetivo

A medida de eficiência deste problema é a soma dos comprimentos dos contentores usados. A solução ótima é uma solução admissível que permita minimizar a soma dos comprimentos dos contentores utilizados.

Assim, o objetivo deste problema é determinar a quantidade de cada tipo de empacotamento a realizar, de modo a minimizar a soma dos comprimentos dos contentores usados.

Variáveis de decisão

As variáveis de decisão são a quantidade de cada tipo de empacotamento que iremos realizar. Ou seja, aquilo que podemos decidir é o número de vezes que iremos realizar um determinado empacotamento. Cada empacotamento é caracterizado pelo comprimento do espaço vazio do contentor, pelo comprimento do item a empacotar nesse espaço (a inserir nesse espaço) e pelo espaço vazio ser um contentor total ou ser o resto de um contentor.

Representação do problema através de um modelo de programação linear

Para representar este problema através de um modelo de programação linear, temos de entender o que serão as restrições lineares e o que será a função objetivo linear do problema.

A função objetivo irá ser o somatório da quantidade de empacotamentos que utilizem um contentor vazio multiplicada pelo comprimento desses contentores. Afinal, sempre que se realizar um empacotamento com um contentor vazio significa que esse contentor foi utilizado, ou seja, a quantidade de empacotamentos que envolvam um contentor vazio de comprimento X é igual a quantidade de contentores usados de comprimento X. Depois ao multiplicar pelo

comprimento desse contentor, e somando tudo, iremos obter a soma dos comprimentos dos contentores usados, aquilo que queríamos minimizar.

As restrições serão de 3 tipos.

Em primeiro lugar, temos as restrições sobre a quantidade máxima de contentores que podem ser usados. Sabemos que podemos no máximo usar 5 contentores de comprimento 7 e 5 de comprimento 10. Assim, iremos restringir o número de empacotamentos que envolvam contentores vazios de comprimento 7 para no máximo 5, e o número de empacotamentos que envolvam contentores vazios de comprimento 10 para no máximo 5.

Em segundo lugar, temos as restrições sobre a quantidade mínima de itens que iremos empacotar. Sabemos que temos, no mínimo, de empacotar 6 itens de comprimento 2, 10 itens de comprimento 4 e 5 itens de comprimento 5. Assim, iremos restringir o número de empacotamentos que envolvam itens de comprimento 2 para no mínimo 6, o número de empacotamentos que envolvam itens de comprimento 4 para no mínimo 10 e o número de empacotamentos que envolvam itens de comprimento 5 para no mínimo 5.

Em terceiro lugar, temos as restrições que traduzem as regras do funcionamento do sistema. Afinal, apenas podemos empacotar um item num determinado espaço, se esse espaço for um contentor total vazio ou se esse espaço resultar de um empacotamento previamente realizado, ou seja, se o espaço for um resto de contentor, este resto só existe porque foram feitos determinados empacotamentos previamente. Assim, iremos restringir que o número de empacotamentos que envolvam um resto de contentor (um contentor não vazio) é igual ou menor ao número de empacotamentos que podem criar esse resto. Para além de poder ser igual, pode ser menor pois pode existir um resto, mas esse resto pode não ser utilizado para fazer um empacotamento. Por exemplo, um empacotamento em que se insira um item qualquer num espaço de comprimento 9, que é um resto de outros contentores, só poderá ser realizado se previamente se colocou um item de comprimento 2 num contentor vazio de comprimento 11. Assim, o número de empacotamentos que envolvam um espaço vazio de comprimento 9 é menor ou igual ao número de empacotamentos de itens de comprimento 2 num contentor vazio de comprimento 11, uma vez que os empacotamentos de espaço vazio de comprimento 9 dependem diretamente dos empacotamentos de itens de comprimento 2 num contentor vazio de comprimento 11. O que fizemos para os empacotamentos de espaço vazio de comprimento 9, iremos fazer para todos os outros empacotamentos que envolvam um resto de um contentor total, onde ainda se possam inserir outros itens.

Modelo de Programação Linear

Conjuntos

$S = \{11,10,7\} \quad D = \{2,4,5\} \quad R = \{9,8,7,6,5,4,3,2\}$

S: conjunto dos comprimentos dos contentores disponíveis.

D: conjunto dos comprimentos dos itens a empacotar.

R: conjunto dos comprimentos dos restos obtidos por um ou mais empacotamentos, onde ainda podemos inserir mais itens. Assim, estes comprimentos são maiores ou iguais ao comprimento do menor item a empacotar (item de comprimento 2). O resto de comprimento 1 não pertence a este conjunto pois não podemos fazer mais nada com ele, não podemos realizar mais empacotamentos apenas com um resto de comprimento 1.

Variáveis de decisão

Sendo as variáveis de decisão as quantidades de determinados empacotamentos, então as variáveis de decisão são números inteiros não negativos (pertencem a $\mathbb{Z}_{\geq 0}$). Assim, as variáveis de decisão são:

- $cont_{i,j}$ – Número de empacotamentos em contentores vazios de comprimento i, onde se insere um item de comprimento j, com $i \in S, j \in D$ e $cont_{i,j} \in \mathbb{Z}_{\geq 0}$. Várias representações:
  - $cont_{11,5}, cont_{11,4}, cont_{11,2}, cont_{10,5}, cont_{10,4}, cont_{10,2}, cont_{7,5}, cont_{7,4}, cont_{7,2} \in \mathbb{Z}_{\geq 0}$.
  - $cont_{i,j} \in \mathbb{Z}_{\geq 0}, \forall (i \in S \wedge j \in D)$.
- $rest_{i,j}$ – Número de empacotamentos em contentores não vazios, com um espaço vazio de comprimento i, onde se insere um item de comprimento j, com $i \in R$, $j \in D$, $i \geq j$ e $rest_{i,j} \in \mathbb{Z}_{\geq 0}$. Várias representações:
  - $rest_{9,5}, rest_{9,4}, rest_{9,2}, rest_{8,5}, rest_{8,4}, rest_{8,2}, rest_{7,5}, rest_{7,4}, rest_{7,2}, rest_{6,5}, rest_{6,4}, rest_{6,2}, rest_{5,5}, rest_{5,4}, rest_{5,2}, rest_{4,4}, rest_{4,2}, rest_{3,2}, rest_{2,2} \in \mathbb{Z}_{\geq 0}$.
  - $rest_{i,j} \in \mathbb{Z}_{\geq 0}, \forall (i \in R \wedge j \in D \wedge i \geq j)$.

Parâmetros

Os parâmetros deste problema são:

- Comprimento de cada contentor disponível, i ∈ S.
- Comprimento de cada item a empacotar, j ∈ D.
- quantidade disponível de cada contentor de comprimento i ∈ S, q_i.
  - $q_{10} = 5$, $q_7 = 5$ e $q_{11} =$ não limitada.
- quantidade de itens a empacotar de comprimento j ∈ D, c_j.
  - $c_2 = 6$, $c_4 = 10$ e $c_5 = 5$.

Função objetivo

A função objetivo linear z é a soma dos comprimentos dos contentores usados. Neste problema, queremos minimizar a função objetivo (problema de minimização), para assim usarmos menos comprimento de contentores usados.

$$\min z = 11(cont_{11,5} + cont_{11,4} + cont_{11,2}) + 10\big(cont_{10,5} + cont_{10,4} + cont_{10,2}\big) + 7(cont_{7,5} + cont_{7,4} + cont_{7,2})$$

Restrições

As restrições deste modelo de programação linear são:

- $cont_{i,j} \in \mathbb{Z}_{\geq 0}, \forall (i \in S \wedge j \in D) \wedge rest_{i,j} \in \mathbb{Z}_{\geq 0}, \forall (i \in R \wedge j \in D \wedge i \geq j)$: apenas são admissíveis as soluções em que as variáveis de decisão sejam números inteiros e não negativos (restrição de não negatividade).
- $\sum_{j \in D} cont_{10,j} \leq 5$: apenas são admissíveis as soluções que não excedam a quantidade disponível de contentores de comprimento 10;
- $\sum_{j \in D} cont_{7,j} \leq 5$: apenas são admissíveis as soluções que não excedam a quantidade disponível de contentores de comprimento 7;

- $\sum_{i \in S} cont_{i,2} + \sum_{n \in R} rest_{n,2} \geq 6$: apenas são admissíveis as soluções que alcancem a quantidade necessária de itens de comprimento 2;
- $\sum_{i \in S} cont_{i,4} + \sum_{n \in R \cap [4,+\infty]} rest_{n,4} \geq 10$: apenas são admissíveis as soluções que alcancem a quantidade necessária de itens de comprimento 4;
- $\sum_{i \in S} cont_{i,5} + \sum_{n \in R \cap [5,+\infty]} rest_{n,5} \geq 5$: apenas são admissíveis as soluções que alcancem a quantidade necessária de itens de comprimento 5;
- $\sum_{j \in D \cap [0,i]} rest_{i,j} \leq \sum_{k \in S \wedge k-i \in D} cont_{k,k-i} + \sum_{m \in R \wedge m-i \in D} rest_{m,m-i}, \forall (i \in R)$: apenas são admissíveis as soluções que utilizem restos de contentores apenas após obterem esses restos através de outros empacotamentos. Resumidamente, não é admissível utilizar restos que nem sequer existem, que nem sequer foram obtidos previamente.

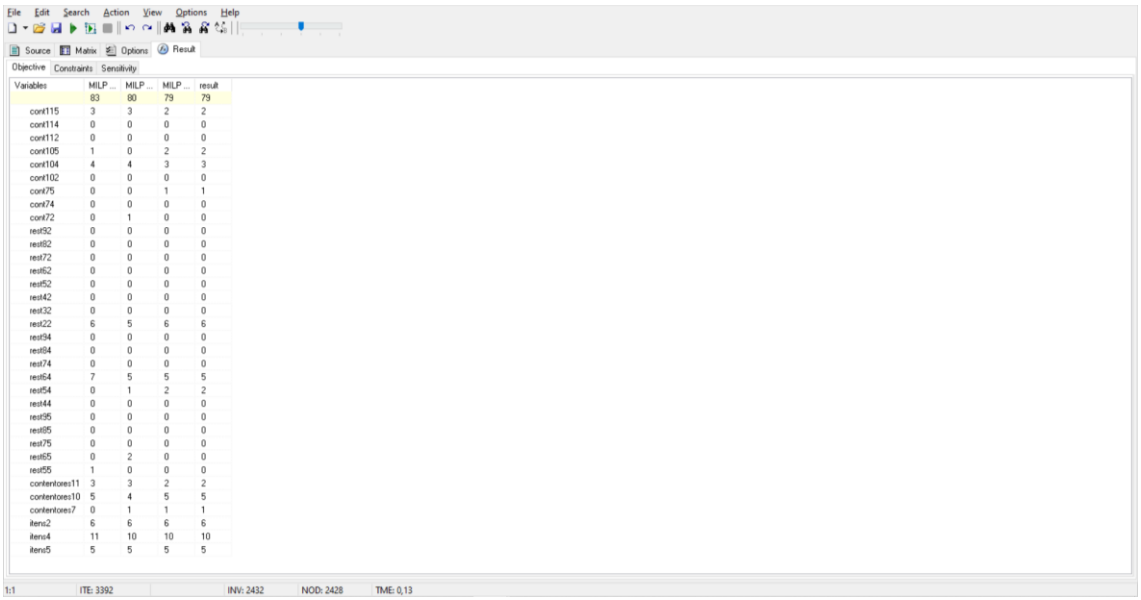
Ficheiro de input

Para resolver este modelo de programação linear usamos o software lpsolve. Neste software, as variáveis de decisão são sempre não negativas, logo não tivemos de incluir a restrição de não negatividade nas restrições do programa (pois já lá está escondida).

Imagem 1 – Print do ficheiro de input do programa lpsolve.

Ficheiro de output

Imagem 2 – Print do ficheiro de output do programa lpsolve.



Interpretação e apresentação da solução ótima

Interpretação da solução ótima

A solução ótima consiste em empacotar:

- 2 vezes um item de comprimento 5 num contentor de comprimento 11 vazio;
- 2 vezes um item de comprimento 5 num contentor de comprimento 10 vazio;
- 3 vezes um item de comprimento 4 num contentor de comprimento 10 vazio;
- 1 vez um item de comprimento 5 num contentor de comprimento 7 vazio;
- 6 vezes um item de comprimento 2 num espaço restante de comprimento 2;
- 5 vezes um item de comprimento 4 num espaço restante de comprimento 6;
- 2 vezes um item de comprimento 4 num espaço restante de comprimento 5.

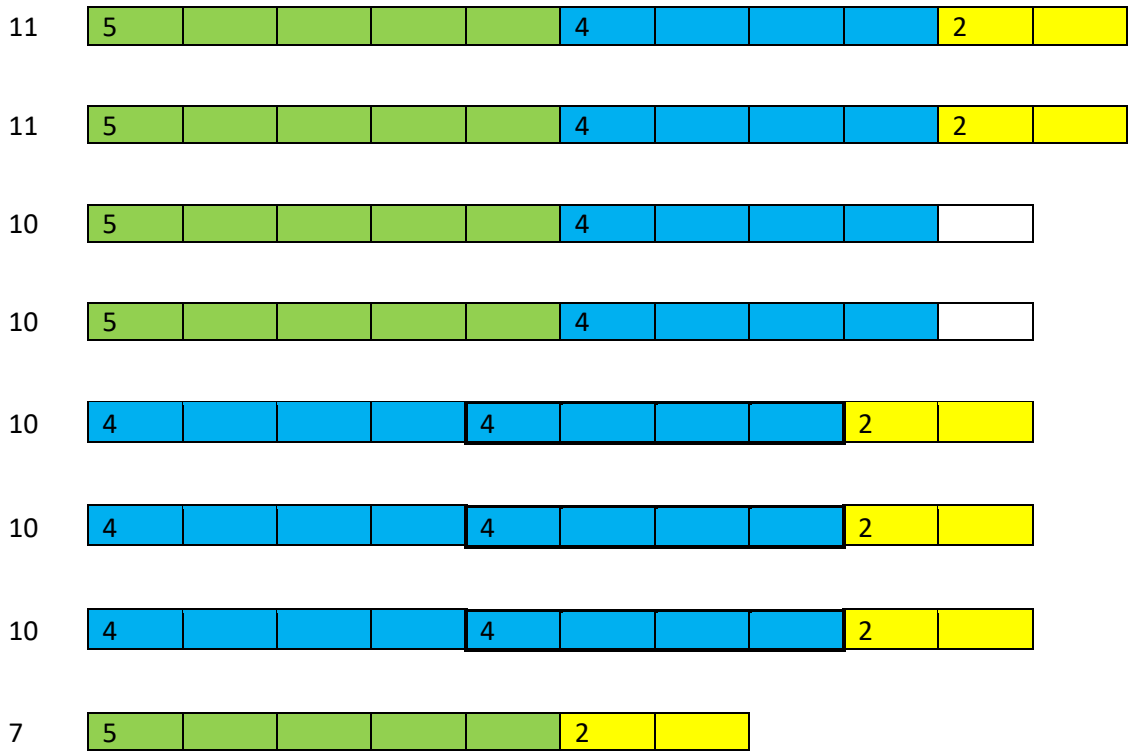
As quatro primeiras operações usam os contentores vazios (2 contentores de comprimento 11, 5 contentores de comprimento 10 e 1 contentor de comprimento 7) e deixam alguns restos. Deixam 5 espaços vazios de comprimento 6, 2 espaços vazios de comprimento 5 e 1 espaço vazio de comprimento 2. Utilizando os 2 espaços vazios de comprimento 5, conseguimos realizar a sétima operação sobrando no fim 2 espaços vazios de comprimento 1. Utilizando os 5 espaços vazios de comprimento 6, conseguimos realizar a sexta operação e deixamos 5 espaços vazios de comprimento 2, para juntar com um que já tínhamos anteriormente. Com 6 espaços vazios de comprimento 2, conseguimos realizar a quinta operação, que não deixa nenhum espaço restante. Observando todas as operações e o número de vezes que foram realizadas, também conseguimos perceber que conseguimos empacotar 6 itens de comprimento 2, 10 itens de comprimento 4 e 5 itens de comprimento 5.

Assim, facilmente percebemos que esta é uma solução admissível do problema, pois conseguimos empacotar todos os itens sem exceder a capacidade dos contentores nem a quantidade disponível de cada um e usando apenas espaços restantes que já existiam, ou seja, conseguimos realizar estas operações no sistema real, respeitando todas as restrições do problema.

Na solução ótima, a **soma dos comprimentos dos contentores usados é 79**, uma vez que o valor da função objetivo é 79. Como vimos anteriormente, a soma dos comprimentos dos itens

a empacotar era 77. Assim, a melhor solução (a solução ótima) aproximou ao máximo soma dos comprimentos dos contentores usados ao valor da soma dos comprimentos dos itens a empacotar, na tentativa de utilizar menos comprimento, mas continuando a respeitar todas as restrições. Ao interpretar a solução ótima e posteriormente na apresentação da mesma, vemos que existem dois espaços vazios de comprimento 1 que não foram usados, o que explica o valor da função objetivo ser 79, uma vez que 77+1+1=77+2=79. O ideal seria a soma dos comprimentos dos contentores usados ser 77, pois 77 era o mínimo que podíamos alcançar (uma vez que era a soma dos comprimentos dos itens a empacotar). No entanto, devido às restrições impostas pelo modelo, o valor ótimo acabou por ser 79.

Apresentação da solução ótima



Aqui temos representada a solução ótima. Temos 2 contentores de comprimento 11, 5 contentores de comprimento de 10 e 1 contentor de comprimento 7. Claramente também temos 6 itens de comprimento 2, 10 itens de comprimento 4 e 5 itens de comprimento 5. Novamente, conseguimos observar que conseguimos empacotar todos os itens sem exceder a capacidade dos contentores nem a quantidade disponível de cada um e usando apenas espaços restantes que já existiam, ou seja, conseguimos realizar estas operações no sistema real, respeitando todas as restrições do problema.

Validação do modelo

Para validar o modelo, iremos verificar se a solução obtida pelo solver é uma decisão admissível e correta do modelo e que pode ser traduzida numa decisão adequada ao sistema real.

Para a solução ser admissível, esta tem de respeitar todas as restrições do modelo, o que realmente acontece como podemos verificar pelos seguintes cálculos:

- $cont_{10,5} + cont_{10,4} + cont_{10,2} \leq 5 \leftrightarrow 2 + 3 + 0 \leq 5 \leftrightarrow 5 \leq 5\ Verdadeiro$
- $cont_{7,5} + cont_{7,4} + cont_{7,2} \leq 5 \leftrightarrow 1 + 0 + 0 \leq 5 \leftrightarrow 1 \leq 5\ Verdadeiro$
- $cont_{11,2} + cont_{10,2} + rest_{9,2} + rest_{8,2} + rest_{7,2} + cont_{7,2} + rest_{6,2} + rest_{5,2} + rest_{4,2} + rest_{3,2} + rest_{2,2} \geq 6 \leftrightarrow 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 6 \geq 6 \leftrightarrow 6 \geq 6\ Verdadeiro$
- $cont_{11,4} + cont_{10,4} + rest_{9,4} + rest_{8,4} + rest_{7,4} + cont_{7,4} + rest_{6,4} + rest_{5,4} + rest_{4,4} \geq 10 \leftrightarrow 0 + 3 + 0 + 0 + 0 + 0 + 5 + 2 + 0 \geq 10 \leftrightarrow 10 \geq 10\ Verdadeiro$
- $cont_{11,5} + cont_{10,5} + rest_{9,5} + rest_{8,5} + rest_{7,5} + cont_{7,5} + rest_{6,5} + rest_{5,5} \geq 5 \leftrightarrow 2 + 2 + 0 + 0 + 0 + 1 + 0 + 0 \geq 5 \leftrightarrow 5 \geq 5\ Verdadeiro$
- $rest_{9,5} + rest_{9,4} + rest_{9,2} \leq cont_{11,2} \leftrightarrow 0 + 0 + 0 \leq 0 \leftrightarrow 0 \leq 0\ Verdadeiro$
- $rest_{8,5} + rest_{8,4} + rest_{8,2} \leq cont_{10,2} \leftrightarrow 0 + 0 + 0 \leq 0 \leftrightarrow 0 \leq 0\ Verdadeiro$
- $rest_{7,5} + rest_{7,4} + rest_{7,2} \leq cont_{11,4} + rest_{9,2} \leftrightarrow 0 + 0 + 0 \leq 0 + 0 \leftrightarrow 0 \leq 0\ Verdadeiro$
- $rest_{6,5} + rest_{6,4} + rest_{6,2} \leq cont_{11,5} + cont_{10,4} + rest_{8,2} \leftrightarrow 0 + 5 + 0 \leq 2 + 3 + 0 \leftrightarrow 5 \leq 5\ Verdadeiro$
- $rest_{5,5} + rest_{5,4} + rest_{5,2} \leq cont_{10,5} + rest_{9,4} + rest_{7,2} + cont_{7,2} \leftrightarrow 0 + 2 + 0 \leq 2 + 0 + 0 + 0 \leftrightarrow 2 \leq 2\ Verdadeiro$
- $rest_{4,4} + rest_{4,2} \leq rest_{9,5} + rest_{8,4} + rest_{6,2} \leftrightarrow 0 + 0 \leq 0 + 0 + 0 \leftrightarrow 0 \leq 0\ Verdadeiro$
- $rest_{3,2} \leq rest_{8,5} + rest_{7,4} + cont_{7,4} + rest_{5,2} \leftrightarrow 0 \leq 0 + 0 + 0 + 0 \leftrightarrow 0 \leq 0\ Verdadeiro$
- $rest_{2,2} \leq rest_{7,5} + cont_{7,5} + rest_{6,4} + rest_{4,2} \leftrightarrow 6 \leq 0 + 1 + 5 + 0 \leftrightarrow 6 \leq 6\ Verdadeiro$

Por fim, iremos calcular a função objetivo para verificar se o solver não cometeu nenhum erro.

- $11 \times (cont_{11,5} + cont_{11,4} + cont_{11,2}) + 10 \times (cont_{10,5} + cont_{10,4} + cont_{10,2}) + 7 \times (cont_{7,5} + cont_{7,4} + cont_{7,2}) = 11 \times (2 + 0 + 0) + 10 \times (2 + 3 + 0) + 7 \times (1 + 0 + 0) = 11 \times 2 + 10 \times 5 + 7 \times 1 = 22 + 50 + 7 = 79$

Como a solução respeita todas as restrições do modelo, então é uma solução admissível. Como o valor da função objetivo dá 79 tanto no solver como à mão, então a solução está correta. Devido aos vários argumentos que demos ao longo deste trabalho, podemos confiar que este modelo retrata com exatidão o problema. Assim, podemos garantir que este modelo é válido.

Conclusão

Para usarmos o modelo de 'úm-corte' de Dyckhoff neste problema, consideramos a operação de empacotar um item num espaço equivalente à operação de corte nos problemas de corte. Inicialmente percebemos que os empacotamentos em espaços vazios que tanto podiam ser um resto como um contentor estavam a dificultar a construção do modelo. Por exemplo, empacotar um item num espaço vazio de comprimento 7, tanto podia ser empacotar um item num contentor vazio de comprimento 7 como podia ser empacotar um item num contentor não vazio de comprimento superior a 7, com um espaço vazio de comprimento 7. Então decidimos considerar dois tipos de empacotamentos, uns que envolviam contentores totais (vazios) e aqueles que envolviam restos de contentores (contentores não vazios). Assim, conseguimos formular o problema, construir um modelo inspirado no modelo de Dyckhoff, colocá-lo num solver, obter a solução ótima, interpretá-la e representá-la e, por fim, validar o modelo.