Universidade do Minho
Escola de Engenharia

[Março de 2025]

Universidade do Minho - Escola de Engenharia

Investigação Operacional - Trabalho Prático 1

Afonso Martins
a106931

Luis Felício
a106913

Gonçalo Castro
a107337

Ricardo Neves
a106850

Francisco Lage
a106813

Sumário

1 Dados do Problema

De acordo com o enunciado, foram calculados os dados do problema relativos ao nosso grupo. Sendo o maior número mecanográfico de aluno entre os membros do grupo o número **107337**, obtiveram-se os seguintes valores:

Comprimento	Quantidade disponível
11	ilimitada
10	8
7	4

Tabela 1: Contentores

Comprimento	Quantidade disponível
1	2
2	11
3	10
4	15
5	5

Tabela 2: Itens

Seguindo as regras definidas no enunciado do trabalho prático:

- $B + 1 = 7 + 1 = 8$, pelo que a quantidade de contentores de comprimento 10 é 8;
- $D + 1 = 3 + 1 = 4$, pelo que a quantidade de contentores de comprimento 7 é 4;
- Dado que $B = 7$ é ímpar, $k_1 = 2$;
- Dado que $C = 3$ é ímpar, $k_2 = 3 + 8 = 11$;
- Dado que $D = 3$ é ímpar, $k_3 = 10$;
- Dado que $E = 7$ é ímpar, $k_4 = 7 + 8 = 15$.

Onde $[k_1..k_4]$ são as quantidades dos itens de comprimento 1 a 4, respetivamente.

A soma dos comprimentos dos itens a empacotar é dada por:

$\sum(\text{quantidade} * \text{comprimento}) = 2 * 1 + 11 * 2 + 10 * 3 + 15 * 4 + 5 * 5 = 139$

2 Formulação do Problema

2.1 Descrição do Problema

O problema apresentado trata-se de um clássico problema de empacotamento, onde se pretende encontrar a forma ótima de empacotar/acomodar uma quantidade de itens de diferentes comprimentos numa quantidade de contentores de, também, diferentes comprimentos. É importante realçar que os itens a empacotar são indivisíveis, isto é, não podem ser divididos de modo a serem acomodados em contentores diferentes.

A quantidade de itens e contentores de cada comprimento para o nosso problema foi especificada no capítulo anterior (Dados do Problema 1).

Vale também realçar que nos foi pedido que a resolução deste problema fosse resolvida respeitando o *Modelo de Arc-Flow*.

O problema é um desafio típico de empacotamento , isto é procurar uma forma de empacotar os itens nos contentores utilizando o *Modelo de Arc-Flow* considerando as suas dimensões, quantidades e a capacidade dos contentores disponíveis e a impossibilidade de dividir os itens.

2.2 Objetivo

O principal objetivo é encontrar a disposição mais eficiente dos itens pelos contentores disponíveis de modo a minimizar o somatório dos comprimentos dos contentores usados.

Alcançar este objetivo requer traduzir o problema para um modelo matemático coerente que represente todas as condições inerentes ao mesmo. Isto só é possível através da definição de um conjunto de variáveis de decisão que permitem formular a função objetivo, que determina o custo da nossa solução, assim como algumas restrições ao problema.

Temos como finalidade encontrar a disposição mais eficiente dos itens dentro dos contentores, de modo que se minimize a a soma dos comprimentos dos contentores usados, isto é, encontrar o mínimo ótimo que otimize a utilização dos contentores em combinação com o comprimento de cada item.

2.3 Definição das Variáveis de Decisão

A definição das variáveis de decisão seguiu a abordagem do **Modelo de Arc-Flow** para a resolução de problemas de empacotamento. Neste modelo, o problema é representado por uma rede de fluxo, onde cada caminho na rede representa uma forma válida de preencher um contentor. Os arcos da rede correspondem às diferentes possibilidades de acomodação dos itens nos contentores, e a solução do problema consiste em encontrar a combinação ótima do fluxos ao longo desses arcos.

No contexto do problema proposto, a rede de fluxo é construída a partir dos comprimentos dos contentores disponíveis e dos itens a serem empacotados. Cada estado na rede representa um determinado preenchimento parcial de um contentor, e cada transição (arco) representa a adição de um item a esse estado. Para garantir uma modelagem eficiente, apenas os estados úteis, onde pelo menos um dos comprimentos resultantes corresponde ao comprimento de um dos itens a empacotar, são considerados.

Os estados úteis do nosso problema são obtidos com base nos conjuntos:

- $S = \{7, 10, 11\}$, que representa os comprimentos dos contentores disponíveis;
- $D = \{1, 2, 3, 4, 5\}$, que representa os comprimentos dos itens a empacotar;
- R, que representa os estados úteis da rede de fluxo, definidos como:

$$R = \left\{\forall_{s \in S, d \in D}\colon\ s - d \mid s - d > 1\right\} = \{10, 9, 8, 7, 6, 5, 4, 3, 2\}$$

Este conjunto de estados úteis garante que padrões inviáveis não sejam considerados, como por exemplo a divisão de um contentor de comprimento 11 em 10/1, onde nenhum dos comprimentos resultantes corresponde a um item válido.

Dessa forma, cada variável de decisão do modelo representa a quantidade de vezes que um determinado arco (padrão de preenchimento) é utilizado na solução final. Esses valores podem ser facilmente interpretados para definir a disposição final dos itens nos contentores.

Por exemplo, uma solução que indique um fluxo correspondente à utilização de um contentor de comprimento 10 preenchido com itens de comprimento 5, 3 e 2 sugere que um caminho válido na rede de fluxo foi seguido, garantindo que os espaços sejam utilizados de forma ótima.

Essa abordagem permite obter soluções ótimas, garantindo que todos os itens sejam empacotados da melhor forma possível nos contentores disponíveis, minimizando o desperdício de espaço e maximizando a eficiência do empacotamento.

2.4 Construção do Modelo

Transformar um problema real para um modelo matemático coerente requer a definição de um conjunto de restrições que permitam definir de forma clara e específica o espaço de soluções admissíveis do mesmo.

No contexto do nosso problema, estas restrições resumem-se essencialmente ao número de contentores disponíveis para cada comprimento e ao número mínimo de itens de cada comprimento que devem ser empacotados. Estas restrições permitem garantir que o modelo nos devolve uma solução onde sejam empacotados todos os itens de diferentes comprimentos e onde não sejam utilizados mais contentores do que aqueles que estão disponíveis. É de notar que poderão existir soluções onde o número de itens acomodados seja superior ao número de itens disponíveis, caso a melhor solução encontrada permita acomodar mais itens sem custos adicionais.

Tendo em conta a forma como foram definidas as variáveis de decisão na seção anterior, os valores a serem restringidos são obtidos através de um somatório específico de quantidades de diferentes padrões de "um-arco". Por exemplo, a quantidade de contentores de um determinado comprimento é dada pela soma das quantidades de padrões de "um-arco" que dele originam, subtraindo eventuais padrões de igual comprimento que originem residualmente de contentores de comprimento superior.

De forma não menos importante, a construção de uma função objetivo adequada é fundamental para assegurar que o modelo desenvolvido trabalha na direção pretendida, minimizando o custo. Neste caso, pretende-se minimizar o comprimento total dos contentores utilizados na solução, pelo que a função objetivo se obtém do somatório da quantidade de cada tipo de contentor que é utilizada, multiplicada pelo comprimento respetivo.

3 Modelação do Problema

3.1 Variáveis de Decisão

- $x_{l,k,x,y}$: número de contentores ou resíduos de comprimento k divididos numa secção de comprimento l e numa secção residual de comprimento $k - l$
- $y_{k,l} \in \mathbb{Z}_{\geq 0}$

  De acordo com os conjuntos S, D e R definidos em Definição das Variáveis de Decisão 2.3:
- $k \in S \cup R = \{11, 10, 9, 8, 7, 6, 5, 4, 3, 2\}$
- $l \in D \land l < k$

A tabela seguinte enumera todas as variáveis resultantes, onde as variáveis destacadas com "-" (hífen) são consideradas **redundantes**, uma vez que representam um padrão de "um-arco" idêntico a uma outra variável, ou simplesmente não fazem sentido no contexto do problema.

		Comprimento do item a inserir (l)					
			1	2	3	4	5
Comprimento do contentor (k)	11	$x_{11,1,11}$	$x_{11,0,1,1}$	$x_{11,0,2,2}$	$x_{11,0,3,3}$	$x_{11,0,4,4}$	$x_{11,0,5,5}$
	11	$x_{11,2,11}$	$x_{11,1,2,1}$	$x_{11,1,3,2}$	$x_{11,1,4,3}$	$x_{11,1,5,4}$	$x_{11,1,6,5}$
	11	$x_{11,3,11}$	$x_{11,2,3,1}$	$x_{11,2,4,2}$	$x_{11,2,5,3}$	$x_{11,2,6,4}$	$x_{11,2,7,5}$
	11	$x_{11,4,11}$	$x_{11,3,4,1}$	$x_{11,3,5,2}$	$x_{11,3,6,3}$	$x_{11,3,7,4}$	$x_{11,3,8,5}$
	11	$x_{11,5,11}$	$x_{11,4,5,1}$	$x_{11,4,6,2}$	$x_{11,4,7,3}$	$x_{11,4,8,4}$	$x_{11,4,9,5}$
	11	$x_{11,6,11}$	$x_{11,5,6,1}$	$x_{11,5,7,2}$	$x_{11,5,8,3}$	$x_{11,5,9,4}$	$x_{11,5,10,5}$
	11	$x_{11,7,11}$	$x_{11,6,7,1}$	$x_{11,6,8,2}$	$x_{11,6,9,3}$	$x_{11,6,10,4}$	$x_{11,6,10,5}$
	11	$x_{11,8,11}$	$x_{11,7,8,1}$	$x_{11,7,9,2}$	$x_{11,7,10,3}$	$x_{11,7,11,4}$	-
	11	$x_{11,9,11}$	$x_{11,8,9,1}$	$x_{11,8,10,2}$	$x_{11,8,11,3}$	-	-
	11	$x_{11,10,11}$	$x_{11,9,10,1}$	$x_{11,9,11,2}$	-	-	-
	11	-	$x_{11,10,11,1}$	-	-	-	-
	10	$x_{10,1,10}$	$x_{10,0,1,1}$	$x_{10,0,2,2}$	$x_{10,0,3,3}$	$x_{10,0,4,4}$	$x_{10,0,5,5}$
	10	$x_{10,2,10}$	$x_{10,1,2,1}$	$x_{10,1,3,2}$	$x_{10,1,4,3}$	$x_{10,1,5,4}$	$x_{10,1,6,5}$
	10	$x_{10,3,10}$	$x_{10,2,3,1}$	$x_{10,2,4,2}$	$x_{10,2,5,3}$	$x_{10,2,6,4}$	$x_{10,2,7,5}$
	10	$x_{10,4,10}$	$x_{10,3,4,1}$	$x_{10,3,5,2}$	$x_{10,3,6,3}$	$x_{10,3,7,4}$	$x_{10,3,8,5}$
	10	$x_{10,5,10}$	$x_{10,4,5,1}$	$x_{10,4,6,2}$	$x_{10,4,7,3}$	$x_{10,4,8,4}$	$x_{10,4,9,5}$
	10	$x_{10,6,10}$	$x_{10,5,6,1}$	$x_{10,5,7,2}$	$x_{10,5,8,3}$	$x_{10,5,9,4}$	$x_{10,5,10,5}$
	10	$x_{10,7,10}$	$x_{10,6,7,1}$	$x_{10,6,8,2}$	$x_{10,6,9,3}$	$x_{10,6,10,4}$	-
	10	$x_{10,8,10}$	$x_{10,7,8,1}$	$x_{10,7,9,2}$	$x_{10,7,10,3}$	-	-
	10	$x_{10,9,10}$	$x_{10,8,9,1}$	$x_{10,8,10,2}$	-	-	-
	10	-	$x_{10,9,10,1}$	-	-	-	-
	7	$x_{7,1,7}$	$x_{7,0,1,1}$	$x_{7,0,2,2}$	$x_{7,0,3,3}$	$x_{7,0,4,4}$	$x_{7,0,5,5}$
	7	$x_{7,2,7}$	$x_{7,1,2,1}$	$x_{7,1,3,2}$	$x_{7,1,4,3}$	$x_{7,1,5,4}$	$x_{7,1,6,5}$
	7	$x_{7,3,7}$	$x_{7,2,3,1}$	$x_{7,2,4,2}$	$x_{7,2,5,3}$	$x_{7,2,6,4}$	$x_{7,2,7,5}$
	7	$x_{7,4,7}$	$x_{7,3,4,1}$	$x_{7,3,5,2}$	$x_{7,3,6,3}$	$x_{7,3,7,4}$	-
	7	$x_{7,5,7}$	$x_{7,4,5,1}$	$x_{7,4,6,2}$	$x_{7,4,7,3}$	-	-
	7	$x_{7,6,7}$	$x_{7,5,6,1}$	$x_{7,5,7,2}$	-	-	-
	7	-	$x_{7,6,7,1}$	-	-	-	-

Figura 1: Declaração de todas as variáveis de decisão

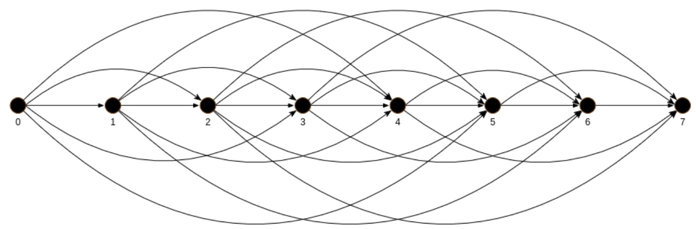

Figura 2: Grafo para o contentor com comprimento 7

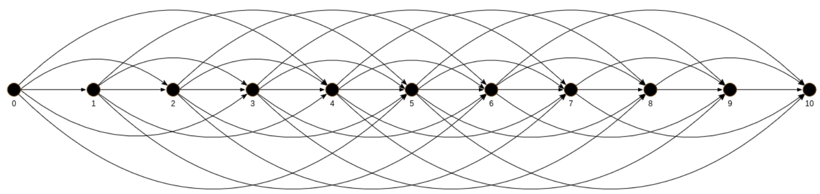

Figura 3: Grafo para o contentor com comprimento 10

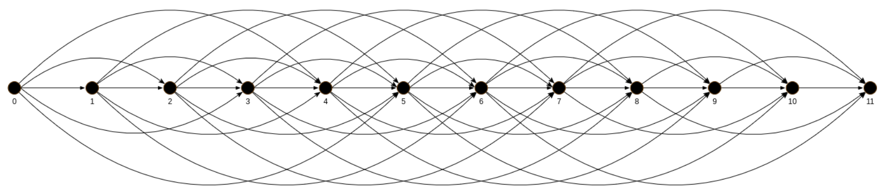

Figura 4: Grafo para o contentor com comprimento 11

3.2 Parâmetros

O problema é constituído por dois parâmetros essenciais:

- Número de contentores de cada comprimento disponíveis;
- Número de itens de cada comprimento a empacotar.

Cujos valores foram já definidos em Dados do Problema 1.

3.3 Função Objetivo

Como mencionado anteriormente, pretende-se **minimizar o comprimento total dos contentores utilizados**. Para isso, basta que a função objetivo componha a soma da quantidade de cada tipo de contentor, multiplicada pelo seu comprimento (que neste problema é o seu custo).

A quantidade de cada tipo de contentor é obtida através da soma de todos os padrões de "um-arco" que dele provêm. No entanto, uma vez que no *Modelo de Arc-Flow* um padrão de "um-arco" como, por exemplo, $y_{7,2}$ pode originar tanto de um contentor de comprimento 7 como de um resíduo de comprimento 7 (*i.e.* a sua origem é indistinguível) é necessário subtrair a quantidade de resíduos de comprimento 7 que são gerados.

Seguindo o exemplo o contentor de comprimento 7, a sua quantidade seria dada por:

$$x_{7_{0,1,1}} + x_{7_{0,2,2}} + x_{7_{0,3,3}} + x_{7_{0,4,4}} + x_{7_{0,5,5}} - x_{10_{1,3,2}} - x_{10_{2,3,1}} - x_{10_{0,3,3}}$$

No entanto, esta abordagem traz outro problema. Na eventualidade de não serem utilizados quaisquer contentores de comprimento 7 mas existirem resíduos desperdiçados do mesmo comprimento, a quantidade será negativa, anulando o custo desse desperdício. Assim, esta fórmula tem de ser adaptada de modo a não admitir valores negativos:

$$\max\Big\{0, x_{7_{0,1,1}} + x_{7_{0,2,2}} + x_{7_{0,3,3}} + x_{7_{0,4,4}} + x_{7_{0,5,5}}\Big\}$$

Realizando este processo de forma análoga para os outros tipos de contentor, obtemos a seguinte função objetivo:

$$(\min)\ 11 * y_{11} + 10 * y_{10} + 7 * y_7$$

Onde,

$$y_7 = \max\Big\{0, x_{7_{0,1,1}} + x_{7_{0,2,2}} + x_{7_{0,3,3}} + x_{7_{0,4,4}} + x_{7_{0,5,5}}\Big\}$$

$$y_{10} = \max\Big\{0, x_{10_{0,1,1}} + x_{10_{0,2,2}} + x_{10_{0,3,3}} + x_{10_{0,4,4}} + x_{10_{0,5,5}}\Big\}$$

$$y_{11} = \max\Big\{0, x_{11_{0,1,1}} + x_{11_{0,2,2}} + x_{11_{0,3,3}} + x_{11_{0,4,4}} + x_{11_{0,5,5}}\Big\}$$

3.4 Restrições

3.4.1 Número de contentores disponíveis

Tal como o processo de construção da função objetivo, também as restrições relativas ao número de contentores disponíveis de cada tipo requerem o cálculo dessa quantidade através das variáveis de decisão definidas.

O cálculo é efetuado de forma análoga, à exceção da necessidade de manter o valor positivo, uma vez que o único objetivo da restrição é impedir que o mesmo ultrapasse um determinado valor. Assim, as restrições são definidas da seguinte forma:

$$x_{7_{0,1,1}} + x_{7_{0,2,2}} + x_{7_{0,3,3}} + x_{7_{0,4,4}} + x_{7_{0,5,5}} \leq 4$$

$$x_{10_{0,1,1}} + x_{10_{0,2,2}} + x_{10_{0,3,3}} + x_{10_{0,4,4}} + x_{10_{0,5,5}} \leq 8$$

Para os contentores de comprimento 7 e 10, respetivamente. Como o contentor de comprimento 11 tem quantidade ilimitada, nenhuma restrição é necessária.

3.4.2 Número de itens a empacotar

Construir restrições que garantam que todos os itens de diferentes comprimentos são empacotados requer, mais uma vez, que essa quantidade seja calculada através das variáveis de decisão definidas.

A quantidade de itens de um dado comprimento é obtida através da soma das quantidades de todos os padrões de "um-arco" que geram pelo menos uma secção de comprimento idêntico ao tipo de item em questão, subtraindo possíveis padrões que efetuem posteriores subdivisões desse comprimento. Assim, a quantidade de, por exemplo, itens de comprimento 5 é dada por:

$$x_{11_{0,5,5}} + x_{11_{1,6,5}} + x_{11_{2,7,5}} + x_{11_{3,8,5}} + x_{11_{4,9,5}} + x_{11_{5,10,5}} + x_{10_{0,5,5}} + x_{10_{1,6,5}} + x_{10_{2,7,5}} + x_{10_{3,8,5}} +$$

$$x_{10_{4,9,5}} + x_{10_{5,10,5}} + x_{7_{0,5,5}} + x_{7_{1,6,5}} + x_{7_{2,7,5}}$$

Realizando o mesmo processo para os restantes comprimentos e aplicando a restrição de quantidade adequada obtemos:

$$x_{11_{0,1,1}} + x_{11_{1,2,1}} + x_{11_{2,3,1}} + x_{11_{3,4,1}} + x_{11_{4,5,1}} + x_{11_{5,6,1}} + x_{11_{6,7,1}} + x_{11_{7,8,1}} + x_{11_{8,9,1}} + x_{11_{9,10,1}} + x_{11_{10,11,1}} + x_{10_{0,1,1}} + x_{10_{1,2,1}} + x_{10_{2,3,1}} + x_{10_{3,4,1}} + x_{10_{4,5,1}} + x_{10_{5,6,1}} + x_{10_{6,7,1}} + x_{10_{7,8,1}} + x_{10_{8,9,1}} + x_{10_{9,10,1}} + x_{7_{0,1,1}} + x_{7_{1,2,1}} + x_{7_{2,3,1}} + x_{7_{3,4,1}} + x_{7_{4,5,1}} + x_{7_{5,6,1}} + x_{7_{6,7,1}} = 2$$

$$x_{11_{0,2,2}} + x_{11_{1,3,2}} + x_{11_{3,5,2}} + x_{11_{4,6,2}} + x_{11_{5,7,2}} + x_{11_{6,8,2}} + x_{11_{7,9,2}} + x_{11_{8,10,2}} + x_{11_{9,11,2}} + x_{10_{0,2,2}} + x_{10_{1,3,2}} + x_{10_{3,5,2}} + x_{10_{4,6,2}} + x_{10_{5,7,2}} + x_{10_{6,8,2}} + x_{10_{7,9,2}} + x_{10_{8,10,2}} + x_{7_{0,2,2}} + x_{7_{1,3,2}} + x_{7_{3,5,2}} + x_{7_{4,6,2}} + x_{7_{5,7,2}} = 11$$

$$x_{11_{0,3,3}} + x_{11_{1,4,3}} + x_{11_{2,5,3}} + x_{11_{3,6,3}} + x_{11_{4,7,3}} + x_{11_{5,8,3}} + x_{11_{6,9,3}} + x_{11_{7,10,3}} + x_{11_{8,11,3}} + x_{10_{0,3,3}} + x_{10_{1,4,3}} + x_{10_{2,5,3}} + x_{10_{3,6,3}} + x_{10_{4,7,3}} + x_{10_{5,8,3}} + x_{10_{6,9,3}} + x_{10_{7,10,3}} + x_{7_{0,3,3}} + x_{7_{1,4,3}} + x_{7_{2,5,3}} + x_{7_{3,6,3}} + x_{7_{4,7,3}} = 10$$

$$x_{11_{0,4,4}} + x_{11_{1,5,4}} + x_{11_{2,6,4}} + x_{11_{3,7,4}} + x_{11_{4,8,4}} + x_{11_{5,9,4}} + x_{11_{6,10,4}} + x_{11_{7,11,4}} + x_{10_{0,4,4}} + x_{10_{1,5,4}} + x_{10_{2,6,4}} + x_{10_{3,7,4}} + x_{10_{4,8,4}} + x_{10_{5,9,4}} + x_{10_{6,10,4}} + x_{7_{0,4,4}} + x_{7_{1,5,4}} + x_{7_{2,6,4}} + x_{7_{3,7,4}} = 15$$

$$x_{11_{0,5,5}} + x_{11_{1,6,5}} + x_{11_{2,7,5}} + x_{11_{3,8,5}} + x_{11_{4,9,5}} + x_{11_{5,10,5}} + x_{10_{0,5,5}} + x_{10_{1,6,5}} + x_{10_{2,7,5}} + x_{10_{3,8,5}} + x_{10_{4,9,5}} + x_{10_{5,10,5}} + x_{7_{0,5,5}} + x_{7_{1,6,5}} + x_{7_{2,7,5}} = 5$$

Para os itens de comprimento 1, 2, 3, 4 e 5, respetivamente.

3.4.3 Restrições adicionais

No processo de obtenção de uma solução, para além de serem gerados espaços de comprimento idêntico a um dos tipos de itens a empacotar, são também gerados comprimentos **intermédios**, isto é, comprimentos residuais que só se provam úteis se forem subdivididos novamente, sendo desperdiçados caso contrário. Este conjunto de comprimentos pode ser definido da seguinte forma, considerando os conjuntos já definidos em Definição das Variáveis de Decisão 2.3:

$$\{x \in R \mid x \notin D)\} = \{6, 8, 9\}$$

É necessário garantir que o modelo desenvolvido não permite a utilização de comprimentos "intermédios" se estes ainda não tiverem sido gerados como resultado de um arco. Para isso, a quantidade de cada um deles deve ser restringida para que nunca assuma valores negativos, o que se obtém de forma análoga ao processo utilizado anteriormente em Número de itens a empacotar 3.4.2, resultando nas seguintes restrições:

$x_{11_{1,6,5}} + x_{11_{2,6,4}} + x_{11_{3,6,3}} + x_{11_{4,6,2}} + x_{11_{5,6,1}} - x_{11_{6,7,1}} - x_{11_{6,8,2}} - x_{11_{6,9,3}} - x_{11_{6,10,4}} + x_{10_{1,6,5}} + x_{10_{2,6,4}} + x_{10_{3,6,3}} + x_{10_{4,6,2}} + x_{10_{5,6,1}} - x_{10_{6,7,1}} - x_{10_{68_2}} - x_{10_{69_3}} - x_{10_{610_4}} + x_{7_{16_5}} + x_{7_{26_4}} + x_{7_{36_3}} + x_{7_{46_2}} + x_{7_{56_1}} - x_{7_{67_1}} \geq 0;$

$$y_{11_3} + y_{10_2} - y_{8_5} - y_{8_4} - y_{8_2} \geq 0$$

$y_{11_2} - y_{9_5} - y_{9_3} - y_{9_2} \geq 0;$

Relativas à quantidade de comprimentos 6, 8 e 9, respetivamente.

4 Resolução do Problema com *LPSolve*

Com o processo de modelação concluído, o problema pode ser inserido no programa *LPSolve* para a sua resolução, efetuando os ajustes de sintaxe necessários para a interpretação do ficheiro pelo programa.

4.1 *Input*

As variáveis de decisão definidas são representadas da seguinte forma no ficheiro de *input* do programa: $y_{l,k,x,y} \Rightarrow$ `yl_k_x_y`.

Segue-se o ficheiro introduzido:

```
min: 11*y_11 + 10*y_10 + 7*y_7;

y_11 - x11_0_1_1 - x11_0_2_2 - x11_0_3_3 - x11_0_4_4 - x11_0_5_5 = 0;
x11_0_1_1 - x11_1_2_1 - x11_1_3_2 - x11_1_4_3 - x11_1_5_4 - x11_1_6_5 - x11_1_11 = 0;
x11_0_2_2 + x11_1_2_1 - x11_2_3_1 - x11_2_4_2 - x11_2_5_3 - x11_2_6_4 - x11_2_7_5 - x11_2_11 = 0;
x11_0_3_3 + x11_1_3_2 + x11_2_3_1 - x11_3_4_1 - x11_3_5_2 - x11_3_6_3 - x11_3_7_4 - x11_3_8_5 - x11_3_11 = 0;
x11_0_4_4 + x11_1_4_3 + x11_2_4_2 + x11_3_4_1 - x11_4_5_1 - x11_4_6_2 - x11_4_7_3 - x11_4_8_4 - x11_4_9_5 - x11_4_11
= 0;
x11_0_5_5 + x11_1_5_4 + x11_2_5_3 + x11_3_5_2 + x11_4_5_1 - x11_5_6_1 - x11_5_7_2 - x11_5_8_3 - x11_5_9_4 -
x11_5_10_5 - x11_5_11 = 0;
x11_1_6_5 + x11_2_6_4 + x11_3_6_3 + x11_4_6_2 + x11_5_6_1 - x11_6_7_1 - x11_6_8_2 - x11_6_9_3 - x11_6_10_4 - x11_6_11
= 0;
x11_2_7_5 + x11_3_7_4 + x11_4_7_3 + x11_5_7_2 + x11_6_7_1 - x11_7_8_1 - x11_7_9_2 - x11_7_10_3 - x11_7_11_4 -
x11_7_11 = 0;
x11_3_8_5 + x11_4_8_4 + x11_5_8_3 + x11_6_8_2 + x11_7_8_1 - x11_8_9_1 - x11_8_10_2 - x11_8_11_3 - x11_8_11 = 0;
x11_4_9_5 + x11_5_9_4 + x11_6_9_3 + x11_7_9_2 + x11_8_9_1 - x11_9_10_1 - x11_9_11_2 - x11_9_11 = 0;
x11_5_10_5 + x11_6_10_4 + x11_7_10_3 + x11_8_10_2 + x11_9_10_1 - x11_10_11_1 - x11_10_11 = 0;
x11_1_11 + x11_2_11 + x11_3_11 + x11_4_11 + x11_5_11 + x11_6_11 + x11_7_11 + x11_8_11 + x11_9_11 + x11_10_11 +
x11_7_11_4 + x11_8_11_3 + x11_9_11_2 + x11_10_11_1 = y_11;

y_10 - x10_0_1_1 - x10_0_2_2 - x10_0_3_3 - x10_0_4_4 - x10_0_5_5 = 0;
x10_0_1_1 - x10_1_2_1 - x10_1_3_2 - x10_1_4_3 - x10_1_5_4 - x10_1_6_5 - x10_1_10 = 0;
x10_0_2_2 + x10_1_2_1 - x10_2_3_1 - x10_2_4_2 - x10_2_5_3 - x10_2_6_4 - x10_2_7_5 - x10_2_10 = 0;
x10_0_3_3 + x10_1_3_2 + x10_2_3_1 - x10_3_4_1 - x10_3_5_2 - x10_3_6_3 - x10_3_7_4 - x10_3_8_5 - x10_3_10 = 0;
x10_0_4_4 + x10_1_4_3 + x10_2_4_2 + x10_3_4_1 - x10_4_5_1 - x10_4_6_2 - x10_4_7_3 - x10_4_8_4 - x10_4_9_5 - x10_4_10
= 0;
x10_0_5_5 + x10_1_5_4 + x10_2_5_3 + x10_3_5_2 + x10_4_5_1 - x10_5_6_1 - x10_5_7_2 - x10_5_8_3 - x10_5_9_4 -
x10_5_10_5 - x10_5_10 = 0;
x10_1_6_5 + x10_2_6_4 + x10_3_6_3 + x10_4_6_2 + x10_5_6_1 - x10_6_7_1 - x10_6_8_2 - x10_6_9_3 - x10_6_10_4 - x10_6_10
= 0;
x10_2_7_5 + x10_3_7_4 + x10_4_7_3 + x10_5_7_2 + x10_6_7_1 - x10_7_8_1 - x10_7_9_2 - x10_7_10_3 - x10_7_10 = 0;
x10_3_8_5 + x10_4_8_4 + x10_5_8_3 + x10_6_8_2 + x10_7_8_1 - x10_8_9_1 - x10_8_10_2 - x10_8_10 = 0;
x10_4_9_5 + x10_5_9_4 + x10_6_9_3 + x10_7_9_2 + x10_8_9_1 - x10_9_10_1 - x10_9_10 = 0;
x10_1_10 + x10_2_10 + x10_3_10 + x10_4_10 + x10_5_10 + x10_6_10 + x10_7_10 + x10_8_10 + x10_9_10 +
x10_5_10_5 + x10_6_10_4 + x10_7_10_3 + x10_8_10_2 + x10_9_10_1 = y_10;

y_7 - x7_0_1_1 - x7_0_2_2 - x7_0_3_3 - x7_0_4_4 - x7_0_5_5 = 0;
x7_0_1_1 - x7_1_2_1 - x7_1_3_2 - x7_1_4_3 - x7_1_5_4 - x7_1_6_5 - x7_1_7 = 0;
x7_0_2_2 + x7_1_2_1 - x7_2_3_1 - x7_2_4_2 - x7_2_5_3 - x7_2_6_4 - x7_2_7_5 - x7_2_7 = 0;
x7_0_3_3 + x7_1_3_2 + x7_2_3_1 - x7_3_4_1 - x7_3_5_2 - x7_3_6_3 - x7_3_7_4 - x7_3_7 = 0;
x7_0_4_4 + x7_1_4_3 + x7_2_4_2 + x7_3_4_1 - x7_4_5_1 - x7_4_6_2 - x7_4_7_3 - x7_4_7 = 0;
x7_0_5_5 + x7_1_5_4 + x7_2_5_3 + x7_3_5_2 + x7_4_5_1 - x7_5_6_1 - x7_5_7_2 - x7_5_7 = 0;
x7_1_6_5 + x7_2_6_4 + x7_3_6_3 + x7_4_6_2 + x7_5_6_1 - x7_6_7_1 - x7_6_7 = 0;
x7_1_7 + x7_2_7 + x7_3_7 + x7_4_7 + x7_5_7 + x7_6_7 +
x7_2_7_5 + x7_3_7_4 + x7_4_7_3 + x7_5_7_2 + x7_6_7_1 = y_7;

x11_0_1_1 + x11_1_2_1 + x11_2_3_1 + x11_3_4_1 + x11_4_5_1 + x11_5_6_1 + x11_6_7_1 + x11_7_8_1 + x11_8_9_1 +
x11_9_10_1 + x11_10_11_1 +
x10_0_1_1 + x10_1_2_1 + x10_2_3_1 + x10_3_4_1 + x10_4_5_1 + x10_5_6_1 + x10_6_7_1 + x10_7_8_1 + x10_8_9_1 +
x10_9_10_1 +
x7_0_1_1 + x7_1_2_1 + x7_2_3_1 + x7_3_4_1 + x7_4_5_1 + x7_5_6_1 + x7_6_7_1 = 2;

x11_0_2_2 + x11_1_3_2 + x11_3_5_2 + x11_4_6_2 + x11_5_7_2 + x11_6_8_2 + x11_7_9_2 + x11_8_10_2 + x11_9_11_2 +
x10_0_2_2 + x10_1_3_2 + x10_3_5_2 + x10_4_6_2 + x10_5_7_2 + x10_6_8_2 + x10_7_9_2 + x10_8_10_2 +
x7_0_2_2 + x7_1_3_2 + x7_3_5_2 + x7_4_6_2 + x7_5_7_2 = 11;

x11_0_3_3 + x11_1_4_3 + x11_2_5_3 + x11_3_6_3 + x11_4_7_3 + x11_5_8_3 + x11_6_9_3 + x11_7_10_3 + x11_8_11_3 +
x10_0_3_3 + x10_1_4_3 + x10_2_5_3 + x10_3_6_3 + x10_4_7_3 + x10_5_8_3 + x10_6_9_3 + x10_7_10_3 +
x7_0_3_3 + x7_1_4_3 + x7_2_5_3 + x7_3_6_3 + x7_4_7_3 = 10;

x11_0_4_4 + x11_1_5_4 + x11_2_6_4 + x11_3_7_4 + x11_4_8_4 + x11_5_9_4 + x11_6_10_4 + x11_7_11_4 +
x10_0_4_4 + x10_1_5_4 + x10_2_6_4 + x10_3_7_4 + x10_4_8_4 + x10_5_9_4 + x10_6_10_4 +
x7_0_4_4 + x7_1_5_4 + x7_2_6_4 + x7_3_7_4 = 15;

x11_0_5_5 + x11_1_6_5 + x11_2_7_5 + x11_3_8_5 + x11_4_9_5 + x11_5_10_5 +
x10_0_5_5 + x10_1_6_5 + x10_2_7_5 + x10_3_8_5 + x10_4_9_5 + x10_5_10_5 +
x7_0_5_5 + x7_1_6_5 + x7_2_7_5 = 5;

y_11 >= 0;
y_10 >= 0;
```

```
y_10 <= 8;
y_7 >= 0;
y_7 <= 4;

int x11_0_1_1, x11_0_2_2, x11_0_3_3, x11_0_4_4, x11_0_5_5;
int x11_1_2_1, x11_1_3_2, x11_1_4_3, x11_1_5_4, x11_1_6_5, x11_1_11;
int x11_2_3_1, x11_2_4_2, x11_2_5_3, x11_2_6_4, x11_2_7_5, x11_2_11;
int x11_3_4_1, x11_3_5_2, x11_3_6_3, x11_3_7_4, x11_3_8_5, x11_3_11;
int x11_4_5_1, x11_4_6_2, x11_4_7_3, x11_4_8_4, x11_4_9_5, x11_4_11;
int x11_5_6_1, x11_5_7_2, x11_5_8_3, x11_5_9_4, x11_5_10_5, x11_5_11;
int x11_6_7_1, x11_6_8_2, x11_6_9_3, x11_6_10_4, x11_6_11;
int x11_7_8_1, x11_7_9_2, x11_7_10_3, x11_7_11_4, x11_7_11;
int x11_8_9_1, x11_8_10_2, x11_8_11_3, x11_8_11;
int x11_9_10_1, x11_9_11_2, x11_9_11;
int x11_10_11_1, x11_10_11;

int x10_0_1_1, x10_0_2_2, x10_0_3_3, x10_0_4_4, x10_0_5_5;
int x10_1_2_1, x10_1_3_2, x10_1_4_3, x10_1_5_4, x10_1_6_5, x10_1_10;
int x10_2_3_1, x10_2_4_2, x10_2_5_3, x10_2_6_4, x10_2_7_5, x10_2_10;
int x10_3_4_1, x10_3_5_2, x10_3_6_3, x10_3_7_4, x10_3_8_5, x10_3_10;
int x10_4_5_1, x10_4_6_2, x10_4_7_3, x10_4_8_4, x10_4_9_5, x10_4_10;
int x10_5_6_1, x10_5_7_2, x10_5_8_3, x10_5_9_4, x10_5_10_5, x10_5_10;
int x10_6_7_1, x10_6_8_2, x10_6_9_3, x10_6_10_4, x10_6_10;
int x10_7_8_1, x10_7_9_2, x10_7_10_3, x10_7_10;
int x10_8_9_1, x10_8_10_2, x10_8_10;
int x10_9_10_1, x10_9_10;

int x7_0_1_1, x7_0_2_2, x7_0_3_3, x7_0_4_4, x7_0_5_5;
int x7_1_2_1, x7_1_3_2, x7_1_4_3, x7_1_5_4, x7_1_6_5, x7_1_7;
int x7_2_3_1, x7_2_4_2, x7_2_5_3, x7_2_6_4, x7_2_7_5, x7_2_7;
int x7_3_4_1, x7_3_5_2, x7_3_6_3, x7_3_7_4, x7_3_7;
int x7_4_5_1, x7_4_6_2, x7_4_7_3, x7_4_7;
int x7_5_6_1, x7_5_7_2, x7_5_7;
int x7_6_7_1, x7_6_7;

int y_11, y_10, y_7;
```

4.2 Output

Executando o *input* acima no IDE *LPSolve* temos:

Variables	MILP ...	MILP ...	result
	141	139	139
y_11	3	5	5
y_10	8	7	7
y_7	4	2	2
x11_0_1_1	1	1	1
x11_0_2_2	2	4	4
x11_0_3_3	0	0	0
x11_0_4_4	0	0	0
x11_0_5_5	0	0	0
x11_1_2_1	1	1	1
x11_1_3_2	0	0	0
x11_1_4_3	0	0	0
x11_1_5_4	0	0	0
x11_1_6_5	0	0	0
x11_1_11	0	0	0
x11_2_3_1	0	0	0
x11_2_4_2	0	0	0
x11_2_5_3	0	0	0
x11_2_6_4	0	0	0
x11_2_7_5	3	5	5
x11_2_11	0	0	0
x11_3_4_1	0	0	0
x11_3_5_2	0	0	0
x11_3_6_3	0	0	0
x11_3_7_4	0	0	0
x11_3_8_5	0	0	0
x11_3_11	0	0	0
x11_4_5_1	0	0	0
x11_4_6_2	0	0	0
x11_4_7_3	0	0	0
x11_4_8_4	0	0	0
x11_4_9_5	0	0	0
x11_4_11	0	0	0
x11_5_6_1	0	0	0
x11_5_7_2	0	0	0
x11_5_8_3	0	0	0

Figura 5: Output parte 1

x11_5_9_4	0	0	0
x11_5_10_5	0	0	0
x11_5_11	0	0	0
x11_6_7_1	0	0	0
x11_6_8_2	0	0	0
x11_6_9_3	0	0	0
x11_6_10_4	0	0	0
x11_6_11	0	0	0
x11_7_8_1	0	0	0
x11_7_9_2	0	0	0
x11_7_10_3	0	0	0
x11_7_11_4	3	5	5
x11_7_11	0	0	0
x11_8_9_1	0	0	0
x11_8_10_2	0	0	0
x11_8_11_3	0	0	0
x11_8_11	0	0	0
x11_9_10_1	0	0	0
x11_9_11_2	0	0	0
x11_9_11	0	0	0
x11_10_11_1	0	0	0
x11_10_11	0	0	0
x10_0_1_1	0	0	0
x10_0_2_2	7	3	3
x10_0_3_3	1	4	4
x10_0_4_4	0	0	0
x10_0_5_5	0	0	0
x10_1_2_1	0	0	0
x10_1_3_2	0	0	0
x10_1_4_3	0	0	0
x10_1_5_4	0	0	0
x10_1_6_5	0	0	0
x10_1_10	0	0	0
x10_2_3_1	0	0	0
x10_2_4_2	0	0	0
x10_2_5_3	1	0	0
x10_2_6_4	4	3	3
x10_2_7_5	2	0	0

Figura 6: Output parte 2

x10_2_10	0	0	0
x10_3_4_1	0	0	0
x10_3_5_2	0	0	0
x10_3_6_3	1	4	4
x10_3_7_4	0	0	0
x10_3_8_5	0	0	0
x10_3_10	0	0	0
x10_4_5_1	0	0	0
x10_4_6_2	0	0	0
x10_4_7_3	0	0	0
x10_4_8_4	0	0	0
x10_4_9_5	0	0	0
x10_4_10	0	0	0
x10_5_6_1	0	0	0
x10_5_7_2	0	0	0
x10_5_8_3	1	0	0
x10_5_9_4	0	0	0
x10_5_10_5	0	0	0
x10_5_10	0	0	0
x10_6_7_1	0	0	0
x10_6_8_2	0	0	0
x10_6_9_3	0	0	0
x10_6_10_4	5	7	7
x10_6_10	0	0	0
x10_7_8_1	0	0	0
x10_7_9_2	0	0	0
x10_7_10_3	2	0	0
x10_7_10	0	0	0
x10_8_9_1	0	0	0
x10_8_10_2	0	0	0
x10_8_10	1	0	0
x10_9_10_1	0	0	0
x10_9_10	0	0	0
x7_0_1_1	0	0	0

Figura 7: Output parte 3

x7_0_2_2	1	2	2
x7_0_3_3	3	0	0
x7_0_4_4	0	0	0
x7_0_5_5	0	0	0
x7_1_2_1	0	0	0
x7_1_3_2	0	0	0
x7_1_4_3	0	0	0
x7_1_5_4	0	0	0
x7_1_6_5	0	0	0
x7_1_7	0	0	0
x7_2_3_1	0	0	0
x7_2_4_2	0	0	0
x7_2_5_3	1	2	2
x7_2_6_4	0	0	0
x7_2_7_5	0	0	0
x7_2_7	0	0	0
x7_3_4_1	0	0	0
x7_3_5_2	0	0	0
x7_3_6_3	0	0	0
x7_3_7_4	3	0	0
x7_3_7	0	0	0
x7_4_5_1	0	0	0
x7_4_6_2	0	0	0
x7_4_7_3	0	0	0
x7_4_7	0	0	0
x7_5_6_1	0	0	0
x7_5_7_2	1	2	2
x7_5_7	0	0	0
x7_6_7_1	0	0	0
x7_6_7	0	0	0

Figura 8: Output parte 4

5 Interpretação da Solução Ótima

A análise dos resultados fornecidos pelo *LPSolve* permite concluir que o valor da função objetivo é de **139 unidades** de comprimento, ou seja, o comprimento total dos diferentes tipos de contentores utilizados na solução ótima é 139. A partir dos valores de y7, y10 e y11, conclui-se que a distribuição dos contentores na solução ótima consiste em 5 contentores de comprimento 11, 7 de comprimento 10 e 2 de comprimento 7.

Além disso, as unidades de comprimento associadas a cada tipo de contentor utilizado podem ser facilmente calculadas ao multiplicar o número de contentores de cada tipo pelo respectivo comprimento, conforme segue:

- Para 5 contentores de **comprimento 11**: (5 x 11 = 55) unidades de comprimento;
- Para 7 contentores de **comprimento 10**: (7 x 10 = 70) unidades de comprimento;
- Para 2 contentores de **comprimento 7**: (2 x 7 = 14) unidades de comprimento.

Portanto, o total de 139 unidades de comprimento é composto por:

- 55 unidades provenientes dos contentores de comprimento 11,
- 70 unidades provenientes dos contentores de comprimento 10,
- 14 unidades provenientes dos contentores de comprimento 7.

A análise das variáveis devolvidas pelo *LPSolve* permite compreender de forma detalhada a distribuição do espaço dentro de cada contentor, evidenciando como os itens foram alocados de forma a otimizar a utilização do comprimento disponível. Para ilustrar esse processo, consideremos o arco de padrão `x11_0_2_2`, cuja interpretação pode ser descrita da seguinte forma:

- O **número 0** representa o estado inicial do contentor, indicando que, no momento da alocação, este se encontra completamente vazio, sem qualquer item previamente armazenado.
- O **número 2** corresponde ao comprimento do item que será colocado no contentor, significando que 2 unidades de comprimento serão ocupadas por esse elemento.
- O **número 2 final** reafirma o espaço que foi efetivamente preenchido, confirmando que, após esta operação, o contentor passa a ter 2 unidades de comprimento ocupadas, restando ainda 9 unidades disponíveis para futuras alocações.

Dando continuidade à análise da solução ótima obtida, verifica-se que o espaço residual de 9 unidades foi posteriormente preenchido através de duas novas operações, representadas pelos seguintes arcos:

O arco `x11_2_7_5` indica que, a partir das 2 unidades já ocupadas, foram alocadas 5 unidades adicionais, elevando o total de espaço utilizado para 7 unidades, deixando ainda um resíduo de 4 unidades livres. O arco `x11_7_11_4` demonstra que as 4 unidades remanescentes foram integralmente preenchidas, concluindo a ocupação total do contentor. Desta forma, é possível observar que a sequência de preenchimento deste contentor de comprimento 11 unidades seguiu a seguinte estrutura: 2/5/4, evidenciando um aproveitamento eficiente do espaço disponível.

Esta análise permite concluir que a solução obtida pelo *LPSolve* não só minimiza o desperdício de espaço como garante uma distribuição equilibrada dos itens dentro dos contentores, reforçando a eficácia do modelo de otimização utilizado.

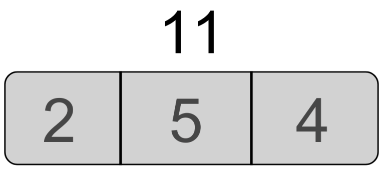

Figura 9: Ilustração da segmentação progressiva de um contentor após a aplicação sucessiva dos 'arcos': `x11_0_2_2, x11_2_7_5` e `x11_7_11_4`

Assim, tendo em conta que existiram dois contentores de 11 divididos como ilustrado na imagem acima, podemos concluir que foi possível armazenar nos mesmos dois itens de comprimento 4, dois itens de comprimento 5 e dois itens de comprimento 2.

Repetindo este mesmo processo para todos os contentores, obtemos a seguinte distribuição:

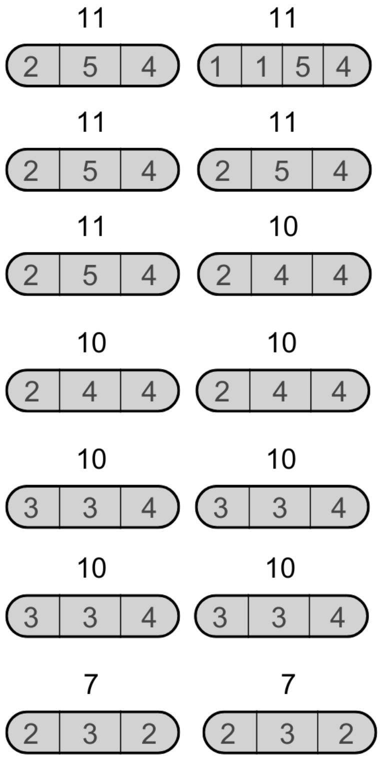

Figura 10: Ilustração da distribuição dos itens pelos contentores

6 Validação do Modelo

Interpretada a solução, é crucial assegurar que a mesma é, de facto, uma solução ótima, que cumpre todas as restrições definidas e que faz sentido na prática.

Numa abordagem inicial, é importante garantir que a solução dada oferece o espaço necessário ao empacotamento dos itens, o que obtemos através da comparação entre o comprimento total dos contentores utilizados e o comprimento total dos itens a empacotar, já calculado em Dados do Problema 1, assim:

$$y_7 * 7 + y_{10} * 10 + y_{11} * 11 \geq 139 \Leftrightarrow$$

$$14 + 70 + 55 \geq 139 \Leftrightarrow$$

$$139 \geq 139$$

Pelo que, em princípio, podemos concluir que existe justamente o espaço necessário para todos os itens. Para além disso, sendo o comprimento dos contentores precisamente igual ao comprimento dos itens e assumindo que a solução é validada como bem definida de seguida, é garantido que a solução é ótima, uma vez que qualquer valor menor implicaria que os itens não coubessem nos contentores.

De seguida, é necessário verificar se o modelo está bem definido, o que implica analisar as variáveis de decisão, a função objetivo e as restrições:

- As variáveis de decisão, conforme definidas em Definição das Variáveis de Decisão 2.3, representam o número de contentores ou resíduos associados a padrões de fluxo (arcos) no Modelo de Arc-Flow, garantindo a conservação do fluxo e a alocação eficiente de itens, conforme o pretendido.
- A função objetivo minimiza o comprimento total dos contentores utilizados, alinhando--se com a estrutura do Arc-Flow.
- As restrições aplicadas aos itens asseguram que cada item é acomodado integralmente, respeitando a sua indivisibilidade e garantindo que todas as unidades são empacotadas.
- As restrições aplicadas aos contentores controlam o número de unidades utilizadas, assegurando que não ultrapassam os limites de disponibilidade definidos.

6.1 Validação das Restrições

De seguida, efetuamos a validação de cada tipo de restrições, de forma a assegurar que, para além de bem definidas, foram cumpridas.

6.1.1 Restrições dos Contentores

$$x_{7_{0,1,1}} + x_{7_{0,2,2}} + x_{7_{0,3,3}} + x_{7_{0,4,4}} + x_{7_{0,5,5}} \leq 4$$

$$0 + 2 + 0 + 0 + 0 \leq 4$$

$$2 \leq 4$$

$$x_{10_{0,1,1}} + x_{10_{0,2,2}} + x_{10_{0,3,3}} + x_{10_{0,4,4}} + x_{10_{0,5,5}} \leq 8$$

$$0 + 4 + 3 + 0 + 0 \leq 8$$

$$7 \leq 8$$

6.1.2 Restrições dos Itens

6.1.2.1 Itens de Comprimento 1

$$x_{11_{0,1,1}} + x_{11_{1,2,1}} + x_{11_{2,3,1}} + x_{11_{3,4,1}} + x_{11_{4,5,1}} + x_{11_{5,6,1}} + x_{11_{6,7,1}} + x_{11_{7,8,1}} + x_{11_{8,9,1}} + x_{11_{9,10,1}} + x_{11_{10,11,1}} + x_{10_{0,1,1}} + x_{10_{1,2,1}} + x_{10_{2,3,1}} + x_{10_{3,4,1}} + x_{10_{4,5,1}} + x_{10_{5,6,1}} + x_{10_{6,7,1}} + x_{10_{7,8,1}} + x_{10_{8,9,1}} + x_{10_{9,10,1}} + x_{7_{0,1,1}} + x_{7_{1,2,1}} + x_{7_{2,3,1}} + x_{7_{3,4,1}} + x_{7_{4,5,1}} + x_{7_{5,6,1}} + x_{7_{6,7,1}} = 2$$

$$1 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 2 \Leftrightarrow 2 = 2$$

6.1.2.2 Itens de Comprimento 2

$$x_{11_{0,2,2}} + x_{11_{1,3,2}} + x_{11_{3,5,2}} + x_{11_{4,6,2}} + x_{11_{5,7,2}} + x_{11_{6,8,2}} + x_{11_{7,9,2}} + x_{11_{8,10,2}} + x_{11_{9,11,2}} + x_{10_{0,2,2}} + x_{10_{1,3,2}} + x_{10_{3,5,2}} + x_{10_{4,6,2}} + x_{10_{5,7,2}} + x_{10_{6,8,2}} + x_{10_{7,9,2}} + x_{10_{8,10,2}} + x_{7_{0,2,2}} + x_{7_{1,3,2}} + x_{7_{3,5,2}} + x_{7_{4,6,2}} + x_{7_{5,7,2}} = 11$$

$$4 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 3 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 2 + 0 + 0 + 0 + 2 = 11 \Leftrightarrow 11 = 11$$

6.1.2.3 Itens de Comprimento 3

$$x_{11_{0,3,3}} + x_{11_{1,4,3}} + x_{11_{2,5,3}} + x_{11_{3,6,3}} + x_{11_{4,7,3}} + x_{11_{5,8,3}} + x_{11_{6,9,3}} + x_{11_{7,10,3}} + x_{11_{8,11,3}} + x_{10_{0,3,3}} + x_{10_{1,4,3}} + x_{10_{2,5,3}} + x_{10_{3,6,3}} + x_{10_{4,7,3}} + x_{10_{5,8,3}} + x_{10_{6,9,3}} + x_{10_{7,10,3}} + x_{7_{0,3,3}} + x_{7_{1,4,3}} + x_{7_{2,5,3}} + x_{7_{3,6,3}} + x_{7_{4,7,3}} = 10$$

$$0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 4 + 0 + 0 + 4 + 0 + 0 + 0 + 0 + 0 + 0 + 2 + 0 + 0 = 10 \Leftrightarrow 10 = 10$$

6.1.2.4 Itens de Comprimento 4

$$x_{11_{0,4,4}} + x_{11_{1,5,4}} + x_{11_{2,6,4}} + x_{11_{3,7,4}} + x_{11_{4,8,4}} + x_{11_{5,9,4}} + x_{11_{6,10,4}} + x_{11_{7,11,4}} + x_{10_{0,4,4}} + x_{10_{1,5,4}} + x_{10_{2,6,4}} + x_{10_{3,7,4}} + x_{10_{4,8,4}} + x_{10_{5,9,4}} + x_{10_{6,10,4}} + x_{7_{0,4,4}} + x_{7_{1,5,4}} + x_{7_{2,6,4}} + x_{7_{3,7,4}} = 15$$

$$0 + 0 + 0 + 0 + 0 + 0 + 0 + 5 + 0 + 0 + 3 + 0 + 0 + 0 + 7 + 0 + 0 + 0 + 0 = 15 \Leftrightarrow 15 = 15$$

6.1.2.5 Itens de Comprimento 5

$$x_{11_{0,5,5}} + x_{11_{1,6,5}} + x_{11_{2,7,5}} + x_{11_{3,8,5}} + x_{11_{4,9,5}} + x_{11_{5,10,5}} + x_{10_{0,5,5}} + x_{10_{1,6,5}} + x_{10_{2,7,5}} + x_{10_{3,8,5}} + x_{10_{4,9,5}} + x_{10_{5,10,5}} + x_{7_{0,5,5}} + x_{7_{1,6,5}} + x_{7_{2,7,5}} = 5$$

$$0 + 0 + 0 + 0 + 5 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 5 \Leftrightarrow 5 = 5$$

6.2 Conclusão da Validação

Após análise detalhada e aplicação das restrições aos contentores e itens, verificamos que os valores obtidos pela solução ótima respeitam todas as condições impostas pelo modelo de programação linear, mais especificamente o *Modelo de Arc-Flow*.

No que diz respeito às restrições dos contentores, observamos que o número de contentores utilizados está dentro dos limites estabelecidos pela disponibilidade, garantindo assim uma utilização eficiente dos recursos.

Para além disso, ao examinar as restrições dos itens, confirmamos que todos os itens foram empacotados adequadamente nos contentores, sem alterações nas suas dimensões originais, atendendo às exigências do problema de empacotamento.

Portanto, com base na solução obtida e na validação das restrições, podemos concluir que o modelo de programação linear desenvolvido é robusto e eficaz para resolver o desafio de empacotamento de itens em contentores, fornecendo uma solução ótima que minimiza o espaço utilizado.

Esta conclusão reforça a confiança na aplicabilidade do modelo proposto e na sua capacidade de oferecer soluções práticas e viáveis para problemas de otimização semelhantes no futuro.